



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

**TEMA: Personalización de un sistema GNU/Linux para administración de
Redes de Área de Almacenamiento.**

Clasificación: Investigación y Desarrollo.

Autores:

Yaniel Arzuaga Ruíz.

Adrian Martinez Casero.

Tutor:

Ing. Anielkis Herrera González.

Co-tutor:

Ing. Mijail Hurtado Fedorovich.

La Habana. Junio de 2011.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad 1 a hacer uso del mismo en su beneficio, siempre y cuando se haga debida referencia a sus autores.

Para que así conste firmamos la presente a los _____ días del mes de _____ de 2011.

Firma del Autor

Yaniel Arzuaga Ruíz.

Firma del Autor

Adrian Martínez Casero.

Firma del Tutor

Ing. Anielkis Herrera González.

Firma del Co-Tutor

Ing. Mijail Hurtado Fedorovich.

Síntesis de los tutores

Síntesis del Tutor: Anielkis Herrera González.

Especialidad de graduación: Ingeniería en Ciencias Informáticas.

Categoría docente: Profesor Instructor.

Años de graduado: 4 años.

Síntesis del Co-Tutor: Mijail Hurtado Fedorovich.

Especialidad de graduación: Ingeniería en Ciencias Informáticas.

Categoría docente: Profesor Instructor.

Años de graduado: 2 años.

¿Has visto hombre solícito en su trabajo? Delante de los reyes estará. No estará delante de los de baja condición.

Prov 22:29

Agradecimientos

Quisiera agradecer a toda mi familia por todo el apoyo que he recibido de ellos, en especial a mi queridísima madre María Elena.

A mi novia Yari por la gran ayuda y todo el apoyo a pesar de lo lejos, eres la mejor.

A mi compañero de tesis Yaniel (El Toro) por contar conmigo para este trabajo.

A todos los que ayudaron, con mucho o poco.

Gracias a Todos.

Adrian.

Le agradezco en primer lugar a mi Dios por darme las fuerzas para hacer esta tesis.

A mi amada esposa por su comprensión y paciencia.

A mis padres por su apoyo.

A mi abuela por su dedicación.

A mi compañero de tesis.

A todos y cada uno de mis hermanos en Cristo de aquí de la UCI.

En fin a todos los que han tenido que ver con la realización de este trabajo.

Yaniel.

Dedicatoria

*Dedico todo mi empeño y esfuerzo puesto en la realización de este trabajo:
A mi familia toda; a los que están y a los que por obra de la Naturaleza no están más
pero siguen en mi mente y mi corazón.*

A mi madre, es lo más grande que tengo sobre la Tierra.

A mi hermano Dariel y mi novia Yari.

Adrian.

*Dedico la realización de este trabajo en primer lugar a Dios que me da la vida y las
fuerzas para hacer todas las cosas, a mi familia que siempre ha estado ahí para
apoyarme por la gracia de Dios, también dedico este trabajo a esta Universidad que
me ha brindado la posibilidad de estudiar en ella y aprender muchas cosas.*

Yaniel.

Resumen

La presente investigación muestra la personalización de un sistema operativo GNU/Linux para administrar de forma eficaz una Red de Área de Almacenamiento. Proceso que incluye la definición de los requisitos que contendrá el sistema, un estudio del estado del arte de las técnicas de almacenamiento más utilizadas. La selección del sistema GNU/Linux al cual se le realizarán los cambios. Además cómo instalar y configurar las nuevas aplicaciones que necesitará la personalización. La implementación de una interfase visual para cada una de las funciones que se agregará para facilitar el trabajo con ellas. Y por último el proceso de pruebas llevado a cabo en un entorno real para garantizar mejor calidad en la solución.

Palabras Claves: Personalización, GNU/Linux, almacenamiento.

Índice de contenido

| | |
|--|----|
| Resumen..... | 7 |
| Introducción..... | 12 |
| Capítulo 1. Técnicas de Almacenamiento..... | 16 |
| Introducción..... | 16 |
| 1.1 Definiciones básicas..... | 16 |
| 1.1.1 Almacenamiento..... | 16 |
| 1.1.2 Software..... | 16 |
| 1.1.3 Hardware..... | 16 |
| 1.1.4 Dispositivo de almacenamiento..... | 16 |
| 1.1.5 Personalizar..... | 16 |
| 1.2 Tecnologías de Almacenamiento de la información..... | 16 |
| 1.2.1 IDE (Integrate Device Electronics) y ATA (Advanced Technology Attachment)..... | 17 |
| 1.2.2 SATA (Serial ATA). | 17 |
| 1.2.3 SCSI (Small Computers System Interface). | 17 |
| 1.3 Tecnología RAID..... | 18 |
| 1.3.1 Niveles de RAID..... | 18 |
| 1.4 Tecnología DAS..... | 21 |
| 1.4.1 Ventajas..... | 21 |
| 1.4.2 Desventajas..... | 22 |
| 1.5 Tecnología NAS..... | 22 |
| 1.5.1 NAS. Ventajas de almacenar en red..... | 22 |
| 1.5.2 Inconvenientes de NAS..... | 23 |
| 1.6 SAN como tecnología de almacenamiento..... | 23 |
| 1.6.1 ¿Por qué el uso de una SAN?..... | 24 |
| 1.6.2 SAN. Características..... | 24 |
| 1.6.3 Ventajas del uso de la SAN..... | 26 |
| 1.6.4 Desventajas..... | 26 |
| 1.7 Estudio de los sistemas operativos para la administración de SAN..... | 26 |
| 1.7.1 Sistemas Privativos..... | 27 |
| 1.7.2 Sistemas Open Source..... | 27 |
| 1.7.3 RedHat Enterprise Linux 6. | 27 |
| 1.7.4 OpenSolaris. | 28 |
| 1.7.5 Openfiler. | 29 |
| 1.7.5.1 Principales características de Openfiler..... | 29 |
| 1.7.5.2 Openfiler en la administración de SAN..... | 29 |
| 1.8 Selección del sistema operativo a personalizar..... | 30 |

| | |
|--|----|
| Conclusiones..... | 30 |
| Capítulo 2. Herramientas en la personalización de Openfiler..... | 31 |
| Introducción..... | 31 |
| 2.1 Lenguajes de Programación..... | 31 |
| 2.1.1 PHP..... | 31 |
| 2.1.2 Bash (Bourne Again Shell). | 32 |
| 2.1.3 JavaScript..... | 32 |
| 2.1.4 HTML..... | 33 |
| 2.2 Entorno de Desarrollo para la personalización..... | 33 |
| 2.3 Herramienta de Gestión de Proyectos..... | 34 |
| 2.4 Metodología de Desarrollo..... | 34 |
| 2.5 Herramienta CASE..... | 35 |
| 2.6 Funcionalidades a Integrar en Openfiler..... | 36 |
| 2.7 Historias de usuario del proyecto..... | 36 |
| Conclusiones..... | 39 |
| Capítulo 3. Descripción de la Personalización..... | 40 |
| Introducción..... | 40 |
| 3.1 Protocolo de Red ATA over Ethernet (AoE)..... | 40 |
| 3.1.1 Principales paquetes a instalar. | 40 |
| 3.1.2 Gestión de paquetes en sistemas GNU/Linux..... | 41 |
| 3.1.3 Instalación de los paquetes vblade y aoetools..... | 41 |
| 3.2 Interfaz de administración de las nuevas funcionalidades..... | 45 |
| 3.2.1 Interfaz Exportar Volumen..... | 49 |
| 3.2.2 Interfaz para Chequeo de Discos..... | 50 |
| 3.2.3 Interfaz Gestión de Energía..... | 52 |
| 3.3 Pruebas al Sistema..... | 53 |
| Conclusiones..... | 54 |
| Conclusiones Generales..... | 55 |
| Recomendaciones..... | 56 |
| Referencias Bibliográficas..... | 57 |
| Bibliografía..... | 59 |
| Anexos..... | 61 |
| Anexo 1. Entrevista realizada a integrante de la empresa cubana GEDEME..... | 61 |
| Anexo 2. Listado total de los paquetes instalados y sus dependencias..... | 61 |
| Anexo 3. Documentación sugerida para las aplicaciones vblade y aoetools..... | 62 |
| Anexo 4. Diagramas..... | 63 |
| Anexo 5. Fragmentos de código..... | 64 |

Índice de tablas

| | |
|---|----|
| Tabla 1: Niveles de RAID..... | 21 |
| Tabla 2: Historia de usuario 1..... | 37 |
| Tabla 3: Tarea 1. Historia de Usuario 1..... | 37 |
| Tabla 4: Tarea 2. Historia de Usuario 1..... | 37 |
| Tabla 5: Tarea 3. Historia de Usuario 1..... | 38 |
| Tabla 6: Historia de Usuario 2..... | 38 |
| Tabla 7: Tarea 1. Historia de Usuario 2..... | 38 |
| Tabla 8: Historia de Usuario 3..... | 39 |
| Tabla 9: Tarea 1. Historia de Usuario 3..... | 39 |
| Tabla 10: Listado de paquetes a instalar..... | 41 |
| Tabla 11: Escenario 1: Exportar Volumen. | 53 |
| Tabla 12: Escenario 2: Chequear Volumen (Vista Estándar)..... | 53 |
| Tabla 13: Escenario 3: Chequear Volumen (Vista Experto). | 54 |
| Tabla 14: Escenario 4: Gestión de Energía..... | 54 |
| Tabla 15: Listado de Paquetes..... | 62 |

Índice de ilustraciones

| | |
|--|----|
| Ilustración 1: RAID 0..... | 19 |
| Ilustración 2: RAID 1..... | 19 |
| Ilustración 3: RAID 0+1..... | 20 |
| Ilustración 4: RAID 5..... | 20 |
| Ilustración 5: Estructura DAS..... | 21 |
| Ilustración 6: Estructura NAS..... | 22 |
| Ilustración 7: Estructura SAN..... | 23 |
| Ilustración 8: Geany..... | 33 |
| Ilustración 9: Ventana de autenticación..... | 45 |
| Ilustración 10: Sección Services..... | 48 |
| Ilustración 11: Sección Volumes..... | 48 |
| Ilustración 12: Volume Functions..... | 49 |
| Ilustración 13: Sección Block Devices..... | 49 |
| Ilustración 14: Vínculo Exportar Volumen..... | 49 |
| Ilustración 15: Interfaz Exportar Volumen..... | 50 |
| Ilustración 16: Vínculo Chequeo de Disco..... | 50 |
| Ilustración 17: Interfaz de Chequeo de Discos, Vista Estándar..... | 51 |
| Ilustración 18: Interfaz de Chequeo de Discos, Vista Experta..... | 51 |

| | |
|--|----|
| Ilustración 19: Vínculo, Gestión de Energía..... | 52 |
| Ilustración 20: Interfaz de Gestión de Energía..... | 52 |
| Ilustración 21: Diagrama de Dominio..... | 63 |
| Ilustración 22: Diagrama de Despliegue..... | 63 |
| Ilustración 23: Implementación de la interfaz web Exportar Volúmenes..... | 64 |
| Ilustración 24: Implementación de la funcionalidad Exportar Volúmenes..... | 64 |
| Ilustración 25: Implementación de la funcionalidad Chequeo de Discos..... | 64 |
| Ilustración 26: Implementación de la funcionalidad para controlar energía de discos..... | 66 |

Introducción

Una de las necesidades de más importancia en la actualidad es almacenar la información generada en la mayor parte de las áreas de comunicación a nivel mundial. La evolución de Internet, el aumento del conocimiento y el desarrollo mismo de la sociedad han provocado un incremento de la información de forma vertiginosa por lo que es necesaria una mayor capacidad de almacenamiento. La vía más común de conservar datos e información es a través de ordenadores, pero estos muchas veces no suplen la demanda de capacidad.

Para satisfacer esta necesidad una de las soluciones que se proponen son las redes de almacenamiento. Este tipo de redes es muy común encontrarlas en instituciones, centros de enseñanza y grandes empresas, lugares donde es elevado tanto el cúmulo de información como la cantidad de personas que tienen acceso a la misma. Por tanto es necesario garantizar cierto grado de seguridad y disponibilidad a los usuarios.

Muchas tecnologías han aparecido para resolver el problema de la centralización de la información, se puede citar por ejemplo la *Storage Area Network (SAN)* o Red de Área de Almacenamiento.

Una SAN es una red concebida para conectar servidores, matrices de discos y librerías de soporte. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.[1]

Según la SNIA¹, SAN puede definirse de las siguientes formas:

1. El propósito de la SAN es la transmisión de datos entre sistemas de almacenamiento y sistemas de almacenamiento o sistema de almacenamiento y servidores de clientes. La estructura SAN contiene las conexiones físicas de los sistemas de almacenamiento para los clientes y, a continuación los dispositivos de gestión de almacenamiento, servidores y dispositivos de red, sin embargo SAN suele definirse como bloques de E/S (*Entrada/Salida*) proveedor de servicios.
2. El sistema de almacenamiento contiene componentes de almacenamiento, dispositivos, equipos informáticos, aplicaciones de software y dispositivos de red.

Con el uso de estas redes se busca ganar en rapidez para acceder a los datos en disco, lograr redundancia y disponibilidad de los mismos.

Este tipo de red de datos es usada por grandes empresas dedicadas a la comercialización y construcción de software y hardware como IBM, SUN, HP. Para administrar los datos en la

¹ Storage Networking Industry Association (Asociación de Industrias de Almacenamiento y Redes).

SAN es necesario contar con un sistema operativo que gestione todos los servicios necesarios para implementar este tipo de red de almacenamiento. Los sistemas operativos más utilizados para esta actividad son privativos y cada uno posee sus ventajas y desventajas, aunque también se conoce de la existencia de tecnologías libres que gestionan estas redes.

El sistema operativo más usado es Microsoft Windows en alrededor del 90% de los ordenadores del mundo, de ahí que las empresas que comercializan y distribuyen las tecnologías de almacenamiento lo usen también como sistema para administrar su producto. La principal desventaja que esto supone para los países subdesarrollados es su alto costo en el mercado.

Por otra parte los sistemas operativos libres o de código abierto (open source) tienen gran desempeño en la actualidad gracias a la comunidad mundial que brinda soporte a los mismos, y se pueden adaptar a una gran parte de las aplicaciones que usan software propietario. La SAN se ha podido adaptar también a estos sistemas.

En Cuba por razones económicas y políticas relacionadas al bloqueo impuesto por los Estados Unidos hace más de 50 años, se dificulta la adquisición en el mercado de esta tecnología. Basado en lo anteriormente expuesto, la Universidad de las Ciencias Informáticas en conjunto con la empresa cubana GEDEME², ha diseñado una estrategia para hacer con esfuerzos propios, un prototipo SAN, funcionando con software libre, que disminuya los costos y brinde además nuevas funcionalidades en comparación a las existentes.

Para garantizar el correcto funcionamiento de la SAN, la empresa GEDEME no cuenta con el entorno apropiado para su administración, siendo esta la **situación problemática** fundamental.

Esas consideraciones permiten reconocer como **problema científico**: ¿Cómo resolver la necesidad de un entorno apropiado para la administración de SAN basado en tecnologías libres?

Siendo el **objeto de estudio**: Proceso de personalización de sistemas GNU/Linux para la administración de redes de almacenamiento, y el **campo de acción**: Proceso de personalización de un sistema GNU/Linux para la administración de SAN.

Atendiendo a la situación problemática descrita y al problema planteado, así como al objeto de estudio y campo de investigación, la presente tesis se estructura y desarrolla en función del siguiente **objetivo principal**: Personalizar un sistema operativo GNU/Linux para administrar eficazmente los recursos de una SAN.

El objetivo propuesto indujo a formular, como guías para el desarrollo de la investigación, las siguientes **preguntas científicas**:

² GEDEME (Gente De Mérito) Empresa Cubana de producción, ensamblaje, instalación y comercialización de sistemas y medios electrónicos e informáticos.

1. ¿Qué herramientas han sido utilizadas para la administración de SAN en el plano del software libre?
2. ¿Cómo mejorar la solución en función de un mejor aprovechamiento de los recursos?
3. ¿Cómo configurar la SAN en entornos empresariales cubanos?

Para hallar las respuestas correspondientes a estas interrogantes se planificaron y desarrollaron las siguientes **tareas**:

1. Investigación acerca de las herramientas utilizadas en la implementación de SAN en el plano nacional e internacional para obtener las principales características de las mismas.
2. Determinación de las herramientas libres adecuadas para la optimización de los recursos.
3. Investigación acerca de las particularidades generales de las empresas cubanas y de los medios que disponen para ofrecer una mejor personalización del producto.
4. Diseño de una propuesta basada en las tecnologías libres seleccionadas.
5. Investigación acerca del diseño de pruebas de aceptación en el producto final.

La investigación exigió la utilización de **métodos de investigación** empíricos, tales como la observación y la entrevista.

La utilización del método **entrevista** permitió conocer las particularidades de los sistemas de administración de redes de almacenamiento con sus principales características.

La **observación** se utilizó para hacer un análisis y una evaluación de la respuesta del sistema, y así determinar si el sistema cumple con los requerimientos funcionales.

Se completó el aseguramiento metodológico con la utilización de los métodos teóricos, estos fueron: **análisis y síntesis** de todo el material acumulado durante la investigación, para arribar a las conclusiones y criterios expuestos en la tesis.

Histórico-Lógico para realizar un estudio analítico de la trayectoria histórica de las herramientas de administración de SAN teniendo en cuenta su evolución hasta la actualidad.

Se determinó que la Investigación iba a contener tres capítulos de los cuales se da un resumen a continuación:

Capítulo 1 Técnicas de Almacenamiento: Expone los principales conceptos relacionados con el almacenamiento de la información en la actualidad así como la administración de SAN, la realización del estado del arte sobre dicho tema y el estudio de las tecnologías libres sobre la administración de SAN para adaptarlas a las necesidades de esta investigación.

Capítulo 2 Personalización de Openfiler: Trata sobre las herramientas utilizadas para la personalización de Openfiler, la metodología de desarrollo de software y las herramientas utilizadas para el desarrollo. Se aborda además acerca de las nuevas funcionalidades y características a incorporar al sistema y una descripción de las mismas mediante historias de usuarios.

Capítulo 3 Descripción de la personalización: Aborda la descripción del proceso de desarrollo de la personalización. Y además está dedicado a probar las nuevas características de Openfiler. También se exponen las pruebas de aceptación para determinar posibles fallos en las funcionalidades de dicho sistema operativo en un escenario virtual y así corregirlos a tiempo antes de una liberación final del producto.

Capítulo 1. Técnicas de Almacenamiento.

Introducción.

Muchas tecnologías son usadas para solucionar el problema de almacenamiento de grandes volúmenes de información que se generan a diario en las empresas. Con la conceptualización de términos importantes respecto a las técnicas de almacenamiento y el estudio del estado del arte que ofrece este capítulo se logra arribar a una sólida conclusión que permite posteriormente la selección del sistema operativo para proceder a su personalización.

1.1 Definiciones básicas.

1.1.1 Almacenamiento.

Acción y resultado de almacenar mercancías para un fin determinado.[2]

Basado en el almacenamiento informático el término refiere el almacenamiento de ficheros, datos, en los que intervienen hardware y software (dispositivos) informáticos, donde se encuentran principalmente los discos ópticos (CD, DVD), discos duros, disquetes, memorias flash, cintas magnéticas.

1.1.2 Software.

Conjunto de programas y rutinas que permiten al ordenador ejecutar determinadas tareas, así como la documentación correspondiente.[2]

1.1.3 Hardware.

Conjunto de elementos materiales o físicos de un sistema informático.[2]

1.1.4 Dispositivo de almacenamiento.

Un dispositivo es un *mecanismo, aparato o máquina que está preparado para realizar una operación prevista.[2]* Por tanto un dispositivo de almacenamiento es un mecanismo preparado para el almacenamiento de datos e información, en este caso la información sería digital.

1.1.5 Personalizar.

Hacer una cosa según el gusto o necesidades del usuario [2]. Para la presente investigación sería adaptar un sistema operativo a las necesidades del cliente.

1.2 Tecnologías de Almacenamiento de la información.

El almacenamiento de la información digital he tenido una evolución muy rápida en los últimos años. Primeramente se guardaba en cintas magnéticas que es un tipo de

almacenamiento no volátil, consiste en una cubierta magnética sobre una tira delgada plástica y fue el primer tipo de memoria secundaria, luego surge el Floppy Disc o disco flexible que fue desplazado por el CD o disco compacto debido a su poca capacidad de almacenamiento y podía deteriorarse con mucha facilidad.[3]

El DVD con una mayor capacidad se impuso en su momento, llegando a alcanzar grandes niveles de comercialización, desplazando así al CD. Pero el avance de los dispositivos de almacenamiento no se ha detenido ejemplo de esta afirmación es la tecnología USB muy difundida en estos momentos y con un incremento en su capacidad en comparación con anteriores tecnologías.

El disco duro o HDD (*Hard Disk Drive*) es el disco que traen las computadoras en su interior, y al pasar los años su precio se ha reducido y la capacidad se ha multiplicado, por esto se ha convertido en la primera opción de almacenamiento. Estos discos duros tienen varias clasificaciones según su interfase en:

1.2.1 IDE (Integrate Device Electronics) y ATA (Advanced Technology Attachment).

Controla los dispositivos de almacenamiento masivo de datos como los discos duros. En la unidad de disco duro hay ATA consecutivo así como ATA paralelo. El IDE es la interfase más comúnmente usada para las unidades de disco duro. La mayor parte de las placas madre tienen el IDE para conectar la unidad de disco duro. Existen por lo general dos enchufes IDE que se alojan a la placa madre. El otro que está presente puede ser usado para el lector de CD u otra unidad de disco duro.[4]

1.2.2 SATA (Serial ATA).

El más novedoso de los estándares de conexión, utiliza un bus serie para la transmisión de datos. Es una interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como puede ser el disco duro, lectores y grabadores de CD/DVD/BR. Proporciona mayores velocidades, mejor aprovechamiento cuando hay varias unidades, mayor longitud del cable de transmisión de datos y capacidad para conectar unidades al instante, es decir, insertar el dispositivo sin tener que apagar el ordenador o que sufra un cortocircuito. Es más rápido y eficiente que IDE.[4]

1.2.3 SCSI (Small Computers System Interface).

En realidad **SCSI** es un tipo de bus; la **interfaz SCSI**, conocida también como **adaptador host**, adopta la forma de una tarjeta que se inserta en una ranura de la placa base, de la que sale un bus, en el que se pueden conectar varios dispositivos. Este adaptador host es en realidad un puente entre el bus SCSI y el bus de la placa-base. Desde el punto de vista del Sistema, los dispositivos SCSI son muy eficientes. Soportan órdenes del tipo "Rebobina esta cinta" o "Formatea este disco" sin intervención del procesador, con lo que se ahorra tiempo de proceso. Esto es especialmente importante en sistemas multitarea como Unix, Linux, OS/2,

Novell Netware y los sistemas Microsoft, a partir de Windows 95. El bus SCSI es muy flexible, y no solo permite conectar discos, también otros, periféricos, como escáneres, unidades de cinta, CD-ROM, DVDs, etc. Estos dispositivos integran la electrónica necesaria que los independiza del adaptador host, y permite que este ignore las características concretas de cada dispositivo conectado.

Generalmente en las computadoras hay un disco duro, esto trae como inconveniente que si se daña, se pierde toda la información, la solución es tener más de un HDD y los datos que se vayan escribiendo en uno, se escriban en el otro (de ahí surge el término redundancia³). A esta solución se le ha denominado RAID (*Redundant Arrays of Independent Disks*) o Contenedores de Almacenamiento Redundante y el propio hardware se encarga de realizar esta tarea.[5]

1.3 Tecnología RAID.

Los discos duros contienen información muy valiosa y se puede dar el caso de que los datos se corrompan por ejemplo en un banco y se pierdan todos esos datos. Por esta y otras razones se utilizan los llamados RAID de discos duros, estos son de varios tipos en dependencia de la cantidad de datos y las características del tipo de copia que se quiera hacer.

De forma sencilla se puede explicar un RAID como la utilización de varios discos duros conectados entre sí. En los cuales se duplican los datos de manera que si ocurre un fallo en alguno de los discos, otro disco del sistema lo reemplace y siga funcionando. También puede darse el caso de querer recuperar la información de uno de los discos a partir de las copias que existan en los restantes discos.

Al utilizar una solución de RAID, puede ampliar la tolerancia a errores. En una configuración de RAID, parte de la capacidad de almacenamiento físico contiene información redundante sobre los datos almacenados en los discos duros. La información redundante es información de paridad o una copia completa y diferente de los datos. Si se produce un error en uno de los discos o en la ruta de acceso, o si no se puede leer un sector del disco, la información redundante permite regenerar los datos.

De RAID existen varias modalidades solo que se abordarán las principales en el siguiente apartado.

1.3.1 Niveles de RAID.

- RAID-0 es una matriz de discos seccionados. Cada disco está dividido de forma lógica de modo que una "sección" abarca todos los discos de la matriz para crear una única partición lógica. Por ejemplo, en una matriz de seis discos, si un archivo se guarda en toda una

³ Repetición de la información contenida (...) que favorece la captación (...) por parte del receptor.[2]

matriz RAID-0 y la aplicación que está guardando el archivo está almacenada en la unidad D, la matriz RAID-0 distribuye el archivo por la unidad lógica D. En este ejemplo, el archivo abarca los seis discos. Desde una perspectiva de rendimiento, RAID-0 es la tecnología de RAID más eficiente, ya que puede escribir en los seis discos simultáneamente. Cuando todos los discos contienen los datos de la aplicación, se obtiene la máxima eficacia en la utilización de los discos. La desventaja de RAID-0 es su falta de tolerancia a errores.[6]

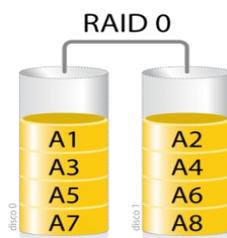


Ilustración 1: RAID 0.

- RAID-1 es una matriz de disco reflejado en la que se reflejan dos discos. RAID-1 es la más confiable de las matrices RAID, puesto que todos los datos se reflejan cuando se escriben. Sólo puede utilizar la mitad del espacio de almacenamiento de los discos. Aunque esto puede resultar poco eficaz, RAID-1 es la opción preferida para los datos que requieren la mayor confiabilidad posible.[6]

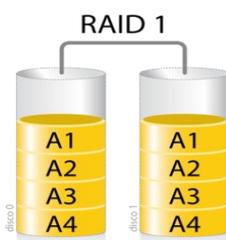


Ilustración 2: RAID 1.

- RAID-0+1. Si el objetivo es lograr una alta confiabilidad y el máximo rendimiento para los datos, es considerable la posibilidad de utilizar RAID-0+1. Este nivel de RAID ofrece alto rendimiento ya que utiliza las ventajas de seccionado de RAID-0, al tiempo que garantiza la redundancia al utilizar las ventajas de reflejo de discos de RAID-1. En una matriz de disco RAID-0+1 los datos se reflejan en los dos conjuntos de disco (RAID-1) y, a continuación, se seccionan entre las unidades (RAID-0). Todos los discos físicos están duplicados en la matriz. Si se tiene una matriz RAID-0+1 de seis discos, tres de los discos estarán disponibles para el almacenamiento de datos.[6]

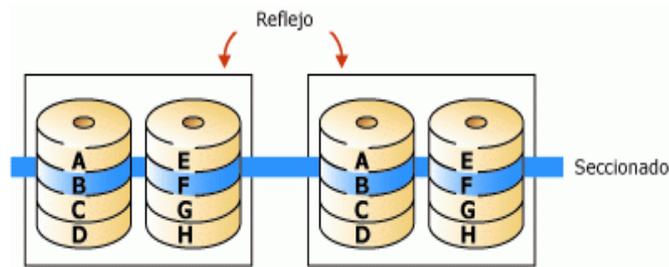


Ilustración 3: RAID 0+1.

- RAID-5 es una matriz de discos seccionados similar a RAID-0 en el sentido de que los datos se distribuyen en toda la matriz. Sin embargo, RAID-5 incluye también paridad. Hay un mecanismo que mantiene la integridad de los datos almacenados en la matriz de forma que, si se producen errores en uno de los discos de la matriz, se puedan reconstruir los datos a partir de los discos restantes. Por tanto, RAID-5 es una solución de almacenamiento confiable. Sin embargo, para mantener la paridad entre los discos, se sacrifica $1/n$ del espacio en disco (donde n es igual al número de unidades de la matriz). Por ejemplo, si se tienen seis discos de 9 GB, tiene 45 GB de espacio de almacenamiento utilizable. Para mantener la paridad, una escritura de datos se traduce en dos escrituras y dos lecturas de la matriz RAID-5. Por tanto, el rendimiento global disminuye. La ventaja de una solución de RAID-5 es que es confiable y utiliza el espacio en disco de forma más eficaz que RAID-1 y RAID-0+1.[6]

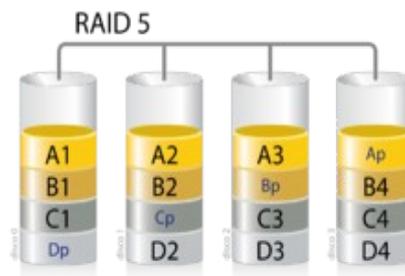


Ilustración 4: RAID 5.

A la hora de diseñar una solución de RAID, hay que tener en cuenta el rendimiento, la confiabilidad, la capacidad de disco y el costo. Si bien el costo y la capacidad son importantes, las principales consideraciones son el rendimiento y la confiabilidad del sistema de almacenamiento.

A continuación se muestra una tabla con la relación costo-confiabilidad de las cuatro soluciones de RAID mencionadas anteriormente, que aunque no son las únicas, si son de las más utilizadas.

| Solución de RAID | Número de Unidades | Costo | Confiabilidad |
|------------------|--------------------|----------|---------------|
| RAID-0 | 10 discos de 9 GB | Alta | Baja |
| RAID-1 | 2 discos de 45 GB | Baja | Baja |
| RAID-0+1 | 20 discos de 9 GB | Muy alta | Muy alta |
| RAID-5 | 11 discos de 9 GB | Alta | Alta |

Tabla 1: Niveles de RAID.

1.4 Tecnología DAS.

Direct Attached Storage, cuya traducción en español sería “Almacenamiento Adjunto Directo”, es considerado el método tradicional de almacenamiento más sencillo. Consiste en conectar el dispositivo de almacenamiento directamente al servidor o estación de trabajo, es decir, físicamente conectado al dispositivo que hace uso de él.

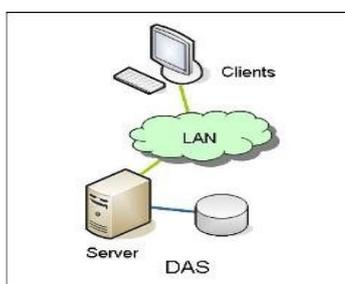


Ilustración 5: Estructura DAS.

En DAS las aplicaciones y programas de usuarios hacen sus peticiones de datos al sistema de ficheros directamente. Dicho sistema de ficheros obtiene los datos requeridos del almacenamiento, además este almacenamiento es local al sistema de ficheros.

¿Qué ventajas y desventajas traería utilizar DAS para el almacenamiento de una red?

1.4.1 Ventajas.

- ✓ Canales de conexión dedicados: la conexión se realiza generalmente por medio de SCSI.
- ✓ Protocolos simples de conexión.
- ✓ Accede a los archivos a bajo nivel, es decir, lo hace como si estuviera accediendo a un sistema de archivos propio.
- ✓ Capacidad: puede utilizar más de 6 discos de una capacidad media de 300Gb cada uno lo que da una capacidad de casi 2 TB, en un espacio bastante reducido.
- ✓ Velocidad y fallos: utiliza discos en RAID, lo que aumenta la velocidad de acceso a los archivos de manera considerable, también provee tolerancia a fallos e integridad.[7]

1.4.2 Desventajas.

Las desventajas de DAS incluyen:

- ✓ Incapacidad para compartir datos o recursos no usados con otros servidores.
- ✓ Distancias limitadas.
- ✓ Recursos reservados de manera exclusiva: la gestión y la actualización de los sistemas DAS como respuesta a los cambios en las aplicaciones resultan especialmente difícil, puesto que sólo admiten la gestión de disco en disco o de servidor en servidor. A los costos de gestión se suman una utilización insuficiente y las pérdidas derivadas.[7]

1.5 Tecnología NAS.

Network Attached Storage, donde su aproximación en español es “Almacenamiento Conectado a la Red” es el nombre dado a una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un computador (Servidor) con ordenadores personales o servidores clientes a través de una red (normalmente TCP/IP), haciendo uso de un sistema operativo optimizado para dar acceso con los protocolos CIFS, NFS, FTP o TFTP.



Ilustración 6: Estructura NAS.

Los protocolos de comunicaciones NAS son basados en ficheros por lo que el cliente solicita el fichero completo al servidor y lo maneja localmente, están por ello orientados a información almacenada en ficheros de pequeño tamaño y gran cantidad.

Muchos sistemas NAS cuentan con más dispositivos de almacenamiento para incrementar su capacidad total. Normalmente, estos dispositivos están dispuestos en RAID.[8][9]

1.5.1 NAS. Ventajas de almacenar en red.

- ✓ Proporciona acceso compartido a ficheros.
- ✓ Escalabilidad. Poseen gran capacidad de almacenamiento, y la posibilidad de insertar

discos en caliente.

- ✓ Servidor especializado con hardware y software optimizado.
- ✓ Protección total para los ficheros. Los dispositivos NAS cuentan con posibilidad para insertar discos adicionales y poder realizar copias de seguridad de forma rápida y automática.
- ✓ Instalación y gestión sencilla. No requiere de mucha experiencia, ya que con sólo conectar el NAS a la corriente y al router es casi suficiente, la gestión es vía web.
- ✓ Utiliza protocolo estándar (NFS, CIFS).
- ✓ Posibilidad de enviar contenido a otros dispositivos como consolas de juego (Xbox, playstation) o smartphones (iphone, etc).
- ✓ Gran compatibilidad con sistemas operativos.

1.5.2 Inconvenientes de NAS.

- Menos rendimiento (comparado con otros sistemas de almacenamiento) debido a que tiene que manejar tanto el tráfico de los usuarios finales y los pedidos de los discos de almacenamiento como las operaciones de respaldo.
- La ausencia de acceso directo a disco debido a que las peticiones de datos se realizan de manera remota.

1.6 SAN como tecnología de almacenamiento.

Una SAN o Red de Área de Almacenamiento, proporciona capacidades de almacenamiento y de administración del almacenamiento para los datos de una empresa. Principalmente, está basada en tecnología Fibre Channel (*Canal de Fibra*) y más recientemente en iSCSI⁴. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.

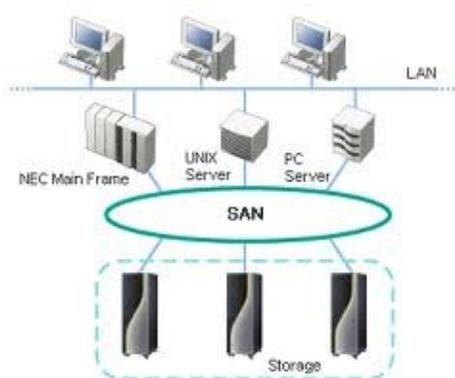


Ilustración 7: Estructura SAN.

⁴ Internet SCSI (Small Computer System interface).

Una red SAN tiene tres capas de componentes principales:

- Capa Host. Esta capa consiste principalmente en Servidores, dispositivos componentes y software (sistemas operativos).
- Capa Fibra. Esta capa la conforman los cables (Fibra óptica) así como los SAN Hubs y los SAN switches como punto central de conexión para la SAN.
- Capa Almacenamiento. Esta capa la componen las formaciones de discos (Disk Arrays, Memoria Caché, RAIDs) y cintas empleados para almacenar datos y protegerlos.

Los proveedores de hardware venden paquetes de SAN completos que incluyen el hardware, el software y el soporte técnico necesarios. El software de SAN administra la redundancia del flujo de datos y de la red al proporcionar varias rutas de acceso a los datos almacenados. Como la tecnología SAN es relativamente nueva y continúa evolucionando rápidamente, puede diseñar e implementar una solución completa de SAN que tenga en cuenta el crecimiento futuro y la aparición de nuevas tecnologías SAN. En última instancia, la tecnología SAN facilita la conectividad entre sistemas de varios proveedores con diferentes sistemas operativos y productos de almacenamiento de distintos proveedores.[10]

1.6.1 ¿Por qué el uso de una SAN?

La necesidad de almacenamiento es evidente, pero no siempre está claro cuál es la solución adecuada para aplicar en una determinada organización. Elegir la solución correcta puede ser una decisión con notables implicaciones, aunque no hay una respuesta correcta única, es necesario centrarse en las necesidades y objetivos finales específicos de cada usuario u organización. Por ejemplo, en el caso concreto de las empresas, el tamaño de la compañía es un parámetro a tener en cuenta y más aun lo es el volumen de información con que trabajan. Para grandes volúmenes de información, una solución SAN sería muy acertada para cumplir con los objetivos de brindar alta seguridad y servicios de accesibilidad y disponibilidad de gran calidad. Por tanto esta es la propuesta a tener en cuenta por parte de los autores del presente trabajo.

1.6.2 SAN. Características.

Latencia - Una de las diferencias y principales características de las SAN es que son construidas para minimizar el tiempo de respuesta del medio de transmisión.

Conectividad - Permite que múltiples servidores sean conectados al mismo grupo de discos o librerías de cintas, permitiendo que la utilización de los sistemas de almacenamiento y los respaldos sean óptimos.

Distancia - Las SAN al ser construidas por lo general con fibra óptica heredan los beneficios de ésta, por ejemplo, pueden tener dispositivos con una separación de hasta 10 Km sin

ruteadores.

Velocidad - El rendimiento de cualquier sistema de computo depende de la velocidad de sus subsistemas, es por ello que las SAN han incrementado su velocidad de transferencia de información, desde 1 Gigabit, hasta actualmente 2 y 4 Gigabits por segundo.

Disponibilidad - Una de las ventajas de las SAN es que al tener mayor conectividad, permiten que los servidores y dispositivos de almacenamiento se conecten más de una vez a la SAN, de esta forma, se pueden tener rutas redundantes que a su vez incrementarán la tolerancia a fallos.

Seguridad - en las SAN esta ha sido desde el principio un factor fundamental. Desde su creación se notó la posibilidad de que un sistema accediera a un dispositivo que no le correspondiera o interfiriera con el flujo de información, es por ello que se ha implementado la tecnología de zonificación, la cual consiste en que un grupo de elementos se aislen del resto para evitar problemas de seguridad, la zonificación puede llevarse a cabo por hardware, software o ambas, siendo capaz de agrupar por puerto o por WWN (*World Wide Name*). Una técnica adicional se implementa a nivel del dispositivo de almacenamiento que es la Presentación, consiste en hacer que una LUN (*Logical Unit Number*) sea accesible sólo por una lista predefinida de servidores o nodos (se implementa con los WWN).

Componentes - Los componentes primarios de una SAN son: switches, directores, HBAs⁵, Servidores, Ruteadores, Gateways, Matrices de discos y Librerías de cintas.

Topología - Cada topología provee distintas capacidades y beneficios las topologías de SAN son:

- Cascada (cascade).
- Anillo (ring).
- Malla (meshed).
- Núcleo/borde (core/edge).

ISL (Inter Switch Link, enlace entre conmutadores) - Actualmente las conexiones entre los switches de SAN se hacen mediante puertos tipo "E" y pueden agruparse para formar una troncal (trunk) que permita mayor flujo de información y tolerancia a fallos.

Arquitectura - channel actuales funcionan bajo dos arquitecturas básicas, FC-AL (*Fibre Channel Arbitrated Loop*) y Switched Fabric, ambos esquemas pueden convivir y ampliar las posibilidades de las SAN. La arquitectura FC-AL puede conectar hasta 127 dispositivos, mientras que switched fabric hasta 16 millones teóricamente.[10]

⁵ Host Bus Adapter (Adaptador de Bus Anfitrión).

1.6.3 Ventajas del uso de la SAN.

- ✓ Si se dispone de varias matrices de discos duros administradas por varios administradores, la administración centralizada de todos los almacenamientos permite que los administradores puedan realizar otras tareas.
- ✓ Es una herramienta valiosa para la administración de datos.
- ✓ Facilitan el establecimiento e implantación de almacenaje repetitivo, como un respaldo en caso de que los servidores fallen.
- ✓ En términos de disponibilidad, ninguna otra solución tiene el potencial para ofrecer la confiabilidad flexible y exhaustiva que una solución de SAN respaldada por un proveedor.
- ✓ Permite agrupar los dispositivos de almacenamiento formando elementos especializados y separados de los servidores. Ya no se necesita una tarjeta RAID y varios discos para cada servidor, con una cabina de discos y varios servidores en una SAN se optimiza la gestión del almacenamiento.
- ✓ La interconexión de todo el almacenamiento dentro de la misma infraestructura de red permite la utilización de las técnicas de gestión globales propias de las redes en una SAN.
- ✓ Permite que varios servidores tengan acceso al mismo volumen de datos por uno o varios caminos dependiendo de la topología y configuración de la misma. Es el escenario adecuado para los entornos críticos y para la implementación de clusters de servidores.
- ✓ Puede extenderse a largas distancias, incluso su tráfico puede ser encaminado a través de otras redes de área extensa. Esto permite soluciones de recuperación ante desastres.[10]

1.6.4 Desventajas.

- ✓ Costo. Una SAN es mucho más costosa que una NAS ya que la primera es una arquitectura completa que utiliza una tecnología que todavía es muy cara. Normalmente, cuando una compañía estima el TCO (*coste total de propiedad*) con respecto al coste por byte, el coste se puede justificar con más facilidad.
- ✓ Asociado al costo. Utiliza como protocolo para las aplicaciones SCSI, que aunque fue originalmente diseñado para trabajar con cables de cobre, su nivel óptimo lo consigue con Fibra Óptica, tecnología muy costosa aun.
- ✓ La capacitación tiene que ser especial para los administradores de estas redes de almacenamiento.

1.7 Estudio de los sistemas operativos para la administración de SAN.

El uso de las redes de almacenamiento por parte de las grandes compañías, instituciones

empresariales y centros de enseñanza superior principalmente, entre otras organizaciones, ha tomado fuerza gracias a las múltiples posibilidades que estas brindan, y uno de los temas más preocupantes es ¿qué sistema operativo utilizar para su administración? Pero si se habla de sistema operativo, muy de la mano va el tipo hardware donde va a estar instalado el sistema en cuestión.

La tecnología SAN fue diseñada con el propósito de no verse limitada ni por hardware ni por el sistema operativo donde vaya a ser instalada, por tanto puede ser soportada por gran variedad de sistemas.

1.7.1 Sistemas Privativos.

Los sistemas operativos privativos son aquellos que no permiten modificaciones en el código fuente de sus programas y hay que pagar licencias para poder utilizarlos. En la administración de redes de almacenamiento se utilizan los siguientes:

- ✓ Microsoft Windows
 - Windows Server 2003, Enterprise Edition.
 - Windows Server 2003, Datacenter Edition.
 - Windows Server 2008.

Entre otras variantes de Windows que también son utilizadas.

1.7.2 Sistemas Open Source.

Los sistemas operativos Open Source son aquellos que permiten modificar el código fuente de sus programas. En la administración de redes de almacenamiento se utilizan los siguientes:

- ✓ GNU/Linux
 - RedHat Enterprise.
 - Openfiler.

Otros Sistemas Open Source.

- OpenSolaris.

Se centrará el estudio principalmente en los sistemas Open Source, dedicados a la administración de Redes de Almacenamiento.

1.7.3 RedHat Enterprise Linux 6.

Es una de las distribuciones Linux más respetadas y renombradas en el mundo de los servidores y en el ambiente empresarial. En conjunto con el software libre que ofrece también brindan uno de los mejores soportes técnicos que existen en la actualidad. Cabe aclarar que en ningún momento venden el código (el cual es muy parecido a su versión libre Fedora) ya que

es libre y gratuito al estar bajo licencia GPL, lo que venden es un soporte técnico profesional.

Dentro de las características más relevantes de esta versión se pueden encontrar cosas interesantes como Linux versión 2.6.32 (con muchas más funciones que la más reciente versión del núcleo de Linux), más núcleos y más memoria (hasta 16 TB de RAM). El nuevo organizador de entrada y salida (I/O scheduler) CFS para optimizar el uso de recursos y consumo de energía. Contiene Ext4 como sistema de ficheros predeterminado (hasta 16TB), nueva interfaz HTTP para gestión y administración de clusters, soporte para dispositivos de almacenamiento SAN de gran tamaño, soporte para utilizar dispositivos iSCSI como particiones de arranque. Posee OpenScap para evaluar parches de seguridad y seguridad en general del sistema, servicio de autenticación a través de SSSD⁶, mejoras en desempeño y funcionalidad de OpenLDAP, Apache 2.2, entre otras características.[11]

1.7.4 OpenSolaris.

Fue un sistema operativo libre publicado en 2005 a partir de la versión privativa de Solaris de Sun Microsystems, ahora parte de Oracle Corporation. OpenSolaris es también el nombre de un proyecto iniciado en 2005 por Sun para construir y desarrollar una comunidad de usuarios alrededor de las tecnologías del sistema operativo del mismo nombre. Después de la adquisición de Sun Microsystems, en agosto de 2010, Oracle decidió interrumpir la publicación y distribución de OpenSolaris, así como su modelo de desarrollo, basado en la disponibilidad de versiones de desarrollo compiladas cada dos semanas y versiones estables cada seis meses. Sin embargo, los términos de su licencia libre no han sido modificados, por lo que el código fuente afectado por ella será publicado cuando Oracle publique nuevas versiones de Solaris.

Este sistema incluye numerosas características que lo convierten en una de las mejores elecciones para soluciones de almacenamiento. Al facilitar el código fuente, Sun ofrece una serie de ventajas: los desarrolladores pueden ahora crear aplicaciones seguras, escalables y fiables de forma más rápida y fácil. Entre las características más relevantes se pueden encontrar, servicios de datos de replicación remota y copia puntual estos servicios crean y administran capturas de datos y proporcionan replicación de datos para migración y recuperación de desastres. OpenSolaris contiene el iniciador y el soporte de destino para el protocolo IETF iSCSI que permite utilizar el protocolo SCSI en redes TCP/IP. Además es uno de los sistemas operativos habilitados para SAN con una pila de canal de fibra completa, incluido transporte de canal de fibra, interfase para hardware y controladores HBA de canal de fibra, APIs de administración de almacenamiento y utilidades de administración de almacenamiento. [12]

⁶ SSSD es un “demonio” (servicio) del sistema. Su función principal es proporcionar acceso a los recursos de identidad y de autenticación a través de un marco común.

1.7.5 Openfiler.

Es una distribución de GNU/Linux que proporciona almacenamiento conectado a la red basado en archivos. Fue creado por Xinit Systems. La última versión disponible del sistema (hasta el momento en que se redacta el presente documento) es la 2.3 y está basada en la distribución de Linux rPath. Es de código abierto y está bajo la licencia GNU General Public License (GPL) versión 2. El código base de Openfiler fue iniciado por Mukund Sivaraman en Xinit Systems en 2001. La compañía creó el proyecto y donó el código base en 2003. En mayo de 2004 se produjo el lanzamiento público de la primera versión de Openfiler.[13]

1.7.5.1 Principales características de Openfiler.

- **Almacenamiento unificado:** Capacidad de Openfiler para centralizar la información en forma de NAS o SAN.
- **Soporte para los protocolos de red:** NFS, SMB/CIFS, HTTP/WebDAV, FTP e iSCSI.
- **El soporte de directorios de red** incluye: NIS, LDAP (con soporte para SMB/CIFS contraseñas encriptadas), Active Directory (en modo nativo y mixto). Los protocolos de autenticación incluyen Kerberos 5.
- **Soporte para particionado:** basado en volúmenes, como sistemas de ficheros nativos en el disco Ext3, JFS y XFS. Posee soporte para múltiples grupos de volúmenes para asignación óptima de almacenamiento.
- **Soporte para:** RAID 0, 1, 5 y 10.
- **Autenticación:** mediante Pluggable Authentication Modules (Módulos de Autenticación Conectables), configurados desde una interfaz web.
- **Instalación en físico o virtual:** muestra su capacidad de adaptación para entornos virtuales sin necesidad de tener una computadora para probar sus funcionalidades.
- **Fuente y Destino iSCSI:** Los llamados initiators (fuente) y targets (destino) que son los servidores y clientes implementando como protocolo de comunicación iSCSI.[13][14][15]

1.7.5.2 Openfiler en la administración de SAN.

Openfiler cubre todos los aspectos claves del almacenamiento de datos:

Fiabilidad: soporta RAID software y hardware con utilidades para monitorización y alertas, instantáneas de volúmenes y recuperación.

Disponibilidad: soporta clusters activo/pasivo de alta disponibilidad, MPIO (soporte de entrada/salida) y replicación a nivel de bloques.

Rendimiento: el kernel de Linux 2.6 soporta las últimas CPU's, hardware de red y almacenamiento.

Escalabilidad: escalabilidad del sistema de archivos mayor de 60TB y soporte para hacer crecer el sistema de archivos y los volúmenes en caliente.

Con las funciones básicas incorporadas puede hacer la función de un servidor NAS basado en sistemas de archivos y de un servidor SAN basado en bloques, todo esto desde el mismo sistema operativo. Además posee una intuitiva interfaz web lo que hace más fácil su manipulación. Openfiler cuenta además con el servicio con igual nombre “openfiler”, encargado de brindar el soporte necesario para la manipulación de los todos sus recursos a través de su interfaz de administración.

Estas últimas características presentadas, la opinión de una inmensidad de usuarios alrededor de todo el orbe acerca de sus acertadas experiencias con este sistema y las probadas experiencias de muchas empresas hacen del mismo un sistema robusto en la administración de redes de almacenamiento.

1.8 Selección del sistema operativo a personalizar.

Teniendo como base el estudio de los apartados anteriores se ha escogido como sistema operativo a personalizar Openfiler, por las múltiples características presentadas anteriormente. Además de permitir la administración de redes SAN y NAS mediante su interfaz web programada en el lenguaje de programación PHP que facilita notablemente esta actividad y la integración de múltiples herramientas. Otra de las razones para utilizar Openfiler es su característica de ser un sistema operativo libre y ser flexible a modificaciones en su código fuente.

Conclusiones.

Se abordaron los principales conceptos asociados al almacenamiento en red y su gestión a través de herramientas libres. Muchas de las definiciones y características anteriormente expuestas servirán como base para el desarrollo del presente trabajo y para la comprensión de los siguientes apartados. Se ha seleccionado Openfiler específicamente para diseñar una personalización del mismo con funcionalidades agregadas de acuerdo a las necesidades expresadas por la empresa GEDEME.

Capítulo 2. Herramientas en la personalización de Openfiler.

Introducción.

El primer aspecto a tener en cuenta para el desarrollo de la solución es determinar el conjunto de herramientas a utilizar para el proceso de construcción de la imagen del sistema. Otro aspecto importante lo conforma los cambios a la interfaz web de Openfiler para lograr incorporar las nuevas características de dicha personalización y la gestión de los programas necesarios que brinden los servicios deseados.

Como principales elementos a tener en cuenta se detectaron:

- Lenguajes de Programación.
- Metodología de Desarrollo.
- Editor de textos para la personalización.
- Instalación y configuración de herramientas que proporcionan nuevas funcionalidades al sistema.

Para realizar las modificaciones previstas es imprescindible hacer uso de estos elementos.

A continuación se describirán cuáles son las herramientas seleccionadas y las consideraciones tenidas en cuenta para su selección.

2.1 Lenguajes de Programación.

Los lenguajes que se mencionan a continuación forman parte de Openfiler y su estudio permitirá realizar las modificaciones necesarias.

2.1.1 PHP.

Es el acrónimo recursivo de PHP Hypertext Preprocessor (en español, Preprocesador Hipertexto PHP). Es un lenguaje de programación diseñado para la creación de páginas web dinámicas, interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. PHP se ha convertido en el lenguaje más utilizado para el desarrollo de aplicaciones del lado del servidor, debido principalmente a su naturaleza open source y su fácil aprendizaje. Es usado para la creación de aplicaciones para servidores, y creación de contenido dinámico para sitios Web. Es un lenguaje de programación gratuito y multiplataforma con una gran librería de funciones y mucha documentación. Presenta compatibilidad con las bases de datos más comunes como MySQL, MS SQL Server, Oracle. Con el mismo se pueden poner en práctica la mayoría de los paradigmas de programación (Ej. Programación Orientada a Objetos), así como

la implementación de patrones de diseño.[16]

Por estas características y además porque es utilizado en la interfaz web de Openfiler es que es seleccionado para darle solución a una parte tan importante del proceso como lo es la interacción usuario-sistema.

2.1.2 Bash (*Bourne Again Shell*).

Es el intérprete de mandatos o “shell” por defecto de muchos de los sistemas UNIX entre ellos Mac OS X y Linux. El shell Bash constituye en este caso la interfaz principal entre el usuario y el sistema, permitiéndole a aquel la interacción con los recursos de éste. El usuario introduce sus órdenes en la terminal⁷, el intérprete las procesa y genera la salida correspondiente. Constituye un lenguaje de programación, que admite crear nuevas órdenes denominadas guiones o “shellscripts”, utilizando combinaciones de mandatos y estructuras lógicas de control, que cuentan con características similares a las del sistema y que permiten que los usuarios y grupos de la máquina cuenten con un entorno personalizado. [17].

De este modo es que se selecciona a Bash como otro de los lenguajes a utilizar pues este se encuentra en Openfiler como intérprete de mandatos por defecto, lo que posibilita una interacción muy cercana con los recursos del mismo. Esto permite mayor compatibilidad y grandes posibilidades para los desarrolladores pues es un lenguaje relativamente fácil de aprender.

2.1.3 JavaScript.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar EMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma del lenguaje del lado del servidor (Server-side JavaScript o SSJS).

Se diseñó con una sintaxis similar al lenguaje C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje de una implementación del Document Object Model (DOM).

JavaScript se interpreta en el agente usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Estas características del lenguaje permitieron su selección en el proceso de

⁷ Dispositivo (generalmente sin interfaz gráfica) que permite la entrada de datos en el ordenador.[2]

implementación de las nuevas funcionalidades de Openfiler. Además JavaScript es muy útil para realizar las validaciones de datos del lado del cliente en aplicaciones web dinámicas.

2.1.4 HTML.

HyperText Markup Language (en español, *Lenguaje de Marcado de Hipertexto*). Es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

También es usado para referirse al contenido del tipo de MIME⁸ text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

Este lenguaje como se explica es el más utilizado para la elaboración de páginas web. Se utiliza para la elaboración de los formularios de contenidos en la interfaz de administración de Openfiler, por lo que para seguir con el mismo estándar se selecciona como otro de los lenguajes a utilizar.

2.2 Entorno de Desarrollo para la personalización.

Geany v0.19.1 es la plataforma de desarrollo utilizada. Es un editor rápido y ligero que permite la edición de varios tipos archivos en diferentes lenguajes incluyendo C, Java, PHP, HTML, Python, Perl, Latex, Pascal, y otros, entre otras valiosas funcionalidades.[18]

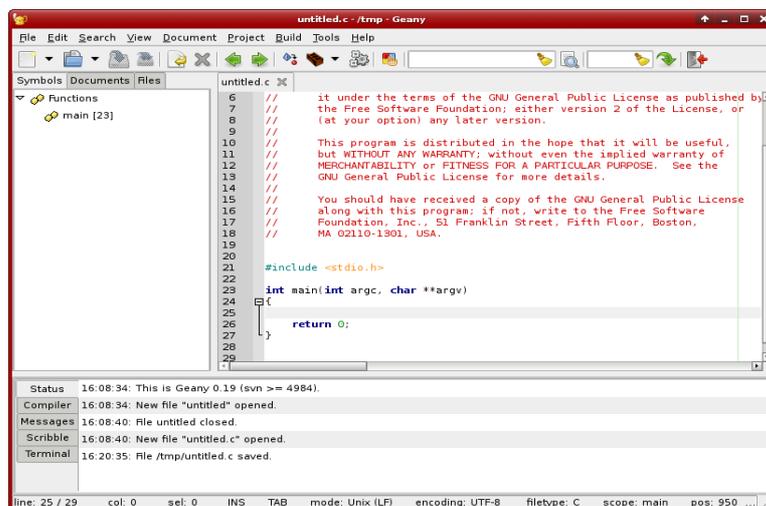


Ilustración 8: Geany.

⁸ Multipurpose Internet Mail Extensions (en español "extensiones multipropósito de correo de internet").

2.3 Herramienta de Gestión de Proyectos.

La gestión de proyectos es una disciplina que se ocupa de la organización y la administración de los recursos en un determinado proyecto. Tiene como principal objetivo poder terminar en tiempo y con la calidad requerida los proyectos propuestos.

En el desarrollo de la personalización de Openfiler, se hace uso de la herramienta RedMine desarrollada en el framework Ruby on Rails. Es una aplicación web con la ventaja de poseer la licencia GPL (GNU General Public License v2).

Entre las principales características de RedMine podemos encontrar las siguientes:

- Soporta múltiples protocolos.
- Publicación de noticias, documentos, archivos.
- Integración con múltiples herramientas.
- Gestión de Riesgos.
- Integración con manejadores de configuración de código por ejemplo SVN (Subversion), CVS y otros.
- Foros.

La integración con la herramienta se realiza mediante la web, por lo que es necesario tener instalado algún navegador web (Mozilla, Opera, Iexplorer).

2.4 Metodología de Desarrollo.

En la personalización de Openfiler se utilizará la metodología de desarrollo de software Extreming Programing (XP) que significa Programación Extrema. Esta se ubica dentro de las llamadas Metodologías Ágiles y se basa en los principios de simplicidad, comunicación y retroalimentación. El proceso de desarrollo de software de esta metodología es iterativo e incremental con iteraciones cortas.

La selección anterior se justifica porque el sistema estará sometido a rápidos cambios lo que traerá consigo algunos riesgos técnicos en la implementación.

Esta metodología contiene 4 fases fundamentales que son **planificación, diseño, desarrollo y pruebas**.^[19]

En XP existen una serie de prácticas, estas pueden expresarse como:

- **El juego de la planificación:** Existe una comunicación frecuente entre el cliente y los programadores.

- **Entregas pequeñas:** Producir rápidamente versiones del sistema que sean operativas aunque no cuenten con todas las funcionalidades del sistema.
- **Metáfora:** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe como debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayuda a nombrar con claridad las clases y métodos del sistema).
- **Diseño Simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- **Refactorización:** Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.[20]
- **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en pareja de programadores.
- **Propiedades colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- **Integración continua:** Cada pieza del código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

Para sacarle el mayor provecho a estas prácticas, es necesario aplicarlas en conjunto porque se apoyan la una en la otra.

La modalidad de programación utilizada en esta metodología es **programación en parejas** esto no significa que dos personas estén programando al mismo tiempo, sino que los dos programadores conocen por donde va cada uno en el desarrollo del trabajo de modo que al ausentarse uno al trabajo, el otro puede continuar sin ningún tipo de problemas. Esta modalidad acelera el desarrollo hacia la solución final y garantiza la menor cantidad de errores y se ajusta perfectamente al modo de trabajo empleado en la investigación. Esta modalidad acelera el desarrollo hacia la solución final y la menor cantidad de errores, esta es otra de las características tenidas en cuenta para escoger XP para el proceso de personalizar la distribución Openfiler.

2.5 Herramienta CASE.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de

vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

2.6 Funcionalidades a Integrar en Openfiler.

La mayor parte de las empresas e instituciones cubanas cuentan con ordenadores de bajas prestaciones, por lo que no pueden hacer un uso eficiente de los recursos. Por tanto es inminente crear alternativas que permitan explotar al máximo las tecnologías. Con el objetivo de un mejor aprovechamiento de los recursos y garantizar un ahorro significativo en costos y energía eléctrica se propone que el nuevo sistema de administración de SAN contendrá entre sus nuevas funcionalidades:

- Exportar volúmenes a través del protocolo de red ATA over Ethernet (AoE).
- Chequeo de discos.
- Control de energía en discos.

Estas funcionalidades se contemplan dentro del proceso de planificación y se describen con mayor especificidad en el apartado siguiente referente a las historias de usuario como técnica utilizada en la metodología XP para especificar los requisitos de software.

2.7 Historias de usuario del proyecto

Estas “ **Historias de usuario** ”, no son más que frases escritas por el cliente donde describe una prestación o proceso que debe cumplir el software.[19] En la presente investigación se expondrán estas historias de usuario conformadas con los requerimientos de la empresa GEDEME. Estas historias de usuario se realizaron tras haber estudiado el proyecto y haber intercambiado con el cliente, para saber sus necesidades.

Historia de usuario 1: Exportar volúmenes usando el protocolo ATA over Ethernet.

| Historia de usuario | |
|--------------------------------------|--|
| No: 1 | Nombre: Exportar volúmenes por AoE. |
| Usuario: Yaniel Arzuaga Ruíz. | |
| Prioridad en Negocio: Alta. | Riesgo en desarrollo: Bajo. |

| | |
|---|------------------------------|
| Puntos estimados: 4/5 | Iteración asignada: 1 |
| Descripción: El usuario escoge la opción export para ir a la sección de exportar volumen. Selecciona el volumen a exportar mediante el protocolo AoE . | |

Tabla 2: Historia de usuario 1.

Para darle solución a la anterior historia de usuario se diseñaron las siguientes tareas:

| Tarea | |
|---|--|
| Número tarea: 1 | Número historia: 1 |
| Nombre tarea: Reunión con clientes. | |
| Tipo de tarea : Desarrollo. | Puntos estimados: 4 |
| Fecha inicio: 1 de octubre de 2010. | Fecha fin: 18 de octubre de 2010. |
| Programador responsable: Equipo de desarrollo. | |
| Descripción: Reunión con los clientes para la selección de las aplicaciones. | |

Tabla 3: Tarea 1. Historia de Usuario 1.

| Tarea | |
|--|---|
| Número tarea: 2 | Número historia: 1 |
| Nombre tarea: Superación en lenguajes de programación PHP y Bash. | |
| Tipo de tarea : Entrenamiento. | Puntos estimados: 3 |
| Fecha inicio: 17 de noviembre de 2010. | Fecha fin: 14 de diciembre de 2010 |
| Programador responsable: Equipo de desarrollo. | |
| Descripción: Estudio de los lenguajes de programación PHP y Bash para el posterior desarrollo de la personalización de Openfiler. | |

Tabla 4: Tarea 2. Historia de Usuario 1.

| Tarea | |
|--|----------------------------|
| Número tarea: 3 | Número historia: 1 |
| Nombre tarea: Activar el Protocolo AoE. | |
| Tipo de tarea : Desarrollo. | Puntos estimados: 4 |

| | |
|---|--|
| Fecha inicio: 21 de diciembre de 2010. | Fecha fin: 21 de enero de 2010. |
| Programador responsable: Equipo de desarrollo. | |
| Descripción: Activar y configurar el módulo de AoE en Openfiler. | |

Tabla 5: Tarea 3. Historia de Usuario 1.

Historia de usuario 2: Chequear discos duros y particiones.

| Historia de usuario | |
|--|---|
| No: 2 | Nombre: Chequear discos duros y particiones. |
| Usuario: Adrian Martínez Casero. | |
| Prioridad en Negocio: Media. | Riesgo en desarrollo: Bajo. |
| Puntos estimados: 3/5 | Iteración asignada: 2 |
| Descripción: El usuario escoge la opción check para ir a la sección de chequear volúmenes y particiones. Selecciona el volumen a chequear. | |

Tabla 6: Historia de Usuario 2.

Para la ejecución de la historia de usuario anterior se definió la tarea siguiente:

| Tarea | |
|---|--|
| Número tarea: 1 | Número historia: 2 |
| Nombre tarea: Chequear discos duros con tecnología S.M.A.R.T. | |
| Tipo de tarea : Desarrollo. | Puntos estimados: 4 |
| Fecha inicio: 1 de marzo de 2011. | Fecha fin: 10 de abril de 2011. |
| Programador responsable: Equipo de desarrollo. | |
| Descripción: Añadir en la interfaz web de Openfiler la opción de chequear los discos duros del sistema para la detección de errores. | |

Tabla 7: Tarea 1. Historia de Usuario 2.

Historia de usuario 3: Controlar energía de discos duros.

| Historia de usuario |
|---------------------|
|---------------------|

| | |
|---|---|
| No: 3 | Nombre: Controlar energía de discos duros. |
| Usuario: Yaniel Arzuaga Ruíz. | |
| Prioridad en Negocio: Media. | Riesgo en desarrollo: Bajo. |
| Puntos estimados: 4/5 | Iteración asignada: 2 |
| Descripción: El usuario escoge la opción de energy para ir a la sección de Gestión de Energía para apagar o encender un disco duro. | |

Tabla 8: Historia de Usuario 3.

Para la ejecución de la historia de usuario anterior se definió la tarea siguiente:

| Tarea | |
|---|---------------------------------|
| Número tarea: 1 | Número historia: 3 |
| Nombre tarea: Controlar energía de discos duros. | |
| Tipo de tarea : Desarrollo | Puntos estimados: 4 |
| Fecha inicio: 30 Marzo 2011 | Fecha fin: 20 Abril 2011 |
| Programador responsable: Equipo de desarrollo | |
| Descripción: Se incorpora en Openfiler la opción de apagado y encendido de discos duros del sistema para el ahorro de energía. | |

Tabla 9: Tarea 1. Historia de Usuario 3

Conclusiones.

A lo largo de todo el capítulo tuvo lugar la selección de un conjunto de herramientas consideradas indispensables para llevar a cabo los pasos para obtener un sistema más robusto en cuanto a sus funcionalidades. La descripción anterior de las historia de usuario así como sus respectivas tareas serán la guía para el proceso de personalización de una herramienta novedosa por incorporar facilidades traducidas a un mejor aprovechamiento de las redes Ethernet con el uso del protocolo AoE, menos costoso que su similar iSCSI que funciona con fibra óptica. Con esto se logrará independencia tecnológica y una forma de ahorrarle recursos al país por concepto de importación de software.

Capítulo 3. Descripción de la Personalización.

Introducción.

Para la personalización de Openfiler se utilizaron distintas herramientas que posibilitaron la incorporación de nuevas funcionalidades. La modificación de la interfaz web y el proceso de pruebas llevadas a cabo también forman parte de este capítulo.

3.1 Protocolo de Red ATA over Ethernet (AoE).

Este protocolo de red creado por la compañía Brantley Coile está diseñado para compartir dispositivos de almacenamiento ATA mediante redes Ethernet. Proporciona la posibilidad de construir redes de almacenamiento de bajo costo con tecnología estándar.

En los sistemas GNU/Linux se puede implementar este protocolo con dos herramientas fundamentales, una es **vblade** y la otra es **aoetools**. La primera se utilizará en el Target o servidor y la segunda en el Initiator o cliente.

La herramienta aoetools posee una serie de utilidades para el trabajo con el protocolo AoE, estas son:

- **aoe-discover**: inicia el proceso de reconocimiento de discos duros ATA a través de la red Ethernet.
- **aoe-interfaces**: restringe los dispositivos de red usados por AoE.
- **aoe-mkdevs**: crea archivos de dispositivos de caracteres y de bloque.
- **aoe-mkshelf**: crea archivos de dispositivos de bloque para una dirección de estante (shelf).
- **aoe-stat**: despliega el estado de los dispositivos AoE.
- **aoeping**: comunicación básica en el espacio del usuario entre los archivos de dispositivos de bloque y los dispositivos AoE.
- **aoe-revalidate**: revalida el tamaño del disco de un dispositivo AoE.

3.1.1 Principales paquetes a instalar.

Para lograr el completo funcionamiento de las herramientas antes mencionadas es necesario instalar los paquetes siguientes y sus dependencias. ([ver Anexo 2](#)).

| Paquetes | Descripción |
|------------|---|
| aoetools | Herramienta para el trabajo con el protocolo AoE. |
| vblade | Herramienta que permite exportar volúmenes ATA mediante redes Ethernet. |
| dosfstools | Herramienta que permite dar formato FAT a dispositivos de almacenamiento. |

Tabla 10: Listado de paquetes a instalar.

3.1.2 Gestión de paquetes en sistemas GNU/Linux.

En los sistemas GNU/Linux los programas se instalan de una forma diferente a como se hace en otros sistemas operativos y una fortaleza considerable de Linux es la gestión de sus paquetes o programas. Esta actividad se realiza mediante una herramienta llamada gestor de paquetes, proporcionando una gran versatilidad a la hora de actualizar todos los paquetes o programas así como la actualización del sistema en general. Ejemplo de estos gestores podemos mencionar, Synaptic en la distribución Debian, Portage en Gentoo, y así cada una de las distribuciones de GNU/Linux que existen poseen su propio gestor de paquetes.

Lamentablemente no siempre se puede usar un gestor de paquetes al necesitar un repositorio⁹ que contenga los paquetes compilados para instalarlos y hacerle la vida más agradable al usuario. En estos casos existen otras alternativas, por ejemplo compilar manualmente el programa deseado en el sistema y configurarlo para trabajar con él. Otra variante sería tomar los paquetes binarios pre-compilados y todas sus dependencias del repositorio de un sistema con iguales características y copiarlos/moverlos hacia el sistema donde finalmente se necesitan.

En el caso particular de este trabajo esta última fue la vía adoptada debido a la deficiencia de no poseer un repositorio de aplicaciones de Openfiler y de un compilador para llevar a cabo el proceso.

Es importante señalar que la estructura original del sistema no sufrirá cambios excepto en los casos de que la instalación de las nuevas aplicaciones así lo requieran.

3.1.3 Instalación de los paquetes vblade y aoetools.

Para lograr que un sistema GNU/Linux cuente con soporte para el protocolo ATA over Ethernet, es necesario instalar en el servidor las aoetools y vblade, y en el cliente solo las aoetools. Como se observaba en los apartados anteriores se explica las diferentes formas de instalación de paquetes en sistemas GNU/Linux teniendo en cuenta esto se procede para este caso.

En primer lugar se deben descargar los paquetes binarios y sus correspondientes

⁹ Almacén, lugar donde se guardan ciertas cosas, en este caso paquetes de programas.[2]

dependencias listadas en el Anexo 1. Para la obtención de dichos paquetes se utilizó el repositorio de Ubuntu (<http://cu.archive.ubuntu.com/ubuntu>). Los paquetes se instalarán en el orden necesario de las dependencias, de forma tal que cuando se instale un paquete ya encuentren instaladas todas sus dependencias.

Para mantener organizada el área de trabajo se confeccionó una estructura siguiendo las siguientes normas:

- Cada aplicación en conjunto con sus dependencias se guardarán en un directorio con igual nombre, por ejemplo, `vblade` se almacenará en el directorio `/vblade` y así sucesivamente con cada programa.
- Cada uno de los directorios correspondientes a las aplicaciones y sus dependencias se almacenarán en el directorio `/trabajo` que a su vez estará contenido en el directorio `/tmp` del sistema.
- Y por último para los archivos descomprimidos se creará el directorio con igual nombre en `/tmp/trabajo/descomprimidos`.

El procedimiento completo para el caso del **Servidor** se haría de esta forma en una terminal:

Paso 1. Creando la estructura de trabajo:

```
$ cd /tmp
$ mkdir trabajo
$ mkdir /trabajo/vblade
$ mkdir /trabajo/aoetools
$ mkdir /trabajo/dosfstools
$ mkdir /trabajo/descomprimidos
```

Con esta estructura ya creada se procede a la descompresión de los archivos binarios de esta manera:

Paso 2. Descomprimiendo los archivos binarios de **aoetools** y **dependencias**.

```
$ cd /trabajo/aoetools
$ ar p aoetools_30-3_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
$ ar p libc6_2.12.1-0ubuntu10.2_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
$ ar p lsb-base_4.0-0ubuntu8_all.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
```

Luego se accede al directorio `/tmp/trabajo/vblade` y se aplica la misma técnica:

Paso 3. Descomprimiendo los archivos binarios de **vblade** y **dependencias**.

```
$ cd /tmp/trabajo/vblade
```

```
$ ar p vblade_16-1_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
$ ar p vblade-persist_0.6-1_all.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
$ ar p runit_2.1.1-4ubuntu1_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
```

Paso 4. Descomprimiendo los archivos binarios de **dosfstools**.

```
$ cd /tmp/trabajo/dosfstools
$ ar p dosfstools_2.11-2.1ubuntu1_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
```

Luego de haber hecho esto ya se encuentran en el directorio `/descomprimidos` todos los ficheros y sus respectivas dependencias.

Quedaría copiarlos para el sistema Openfiler, sería:

Paso 5. Copiando aplicaciones para el sistema.

```
$ cp -r /tmp/trabajo/descomprimidos/* /
```

Paso 6. Configuraciones.

En estas condiciones aún no se pueden usar las **aoetools** porque no se ha cargado el módulo denominado **aoe** que viene por defecto en las versiones más recientes del kernel de linux, para lograrlo debe hacerse como administrador del sistema, y se procede poniendo en una terminal:

```
# modprobe aoe
```

Con lo que automáticamente el módulo quedará activo y las **aoetools** disponibles para su uso.

El paquete **vblade** incluye:

`/etc/vblade.conf` <-- archivo de configuración.

`/etc/init.d/vblade` <-- guión de arranque/parada.

Contenido del fichero `/etc/vblade.conf`:

```
# network_device shelf slot file/disk/partition mac[,mac[,mac]]
# eth0 0 0 /dev/sdb 00:11:22:33:44:55
```

La configuración es simple, debe especificar:

- **network_device**, dispositivo de red por el cual servirá la unidad compartida AoE.
- **shelf**, gabinete, recordar que vblade emula un arreglo de gabinetes y posiciones.
- **slot**, posición en el gabinete.
- **file/disk/partition**, archivo/disco/partición, elemento de bloque a servir vía AoE.

➤ **mac[,mac[,mac]]**, lista de control de acceso de MACs permitidas.

Un ejemplo de configuración de vblade con dos particiones se lograría editando el fichero **vblade.conf** contenido en el directorio **/etc** del sistema. Se debe acceder como administrador con cualquier editor de texto, por ejemplo:

```
# nano /etc/vblade.conf:
```

y se incluyen las siguientes líneas:

```
eth0 0 0 /dev/sdb1 00:11:22:33:44:55,00:11:22:33:44:66
```

```
eth0 0 1 /dev/sdb2 00:11:22:33:44:55,00:11:22:33:44:66
```

Ahora se puede arrancar el servidor AoE.

```
# service vblade start
```

De una vez se configura para que arranque al inicio del sistema:

```
# chkconfig vblade on
```

Una forma sencilla de exportar volúmenes también sería escribir directamente en una terminal la siguiente línea:

```
# vblade 0 0 eth0 /dev/sda_No_partición
```

Donde el primer cero corresponde al **shelf**, el segundo cero al **slot**, **eth0** sería la interfaz de red utilizada y **/dev/sda_No_partición** especifica la partición o disco que se va a exportar.

Para más información visitar la documentación sugerida en el Anexo 2.

En el caso del **Ciente** solo es necesario hacer el procedimiento para aoetools. Sería de este modo igualmente en una terminal:

Paso 1. Estructura de trabajo.

```
$ cd /tmp
$ mkdir trabajo
$ mkdir /trabajo/aoetools
$ mkdir /trabajo/descomprimidos
```

Paso 2. Descomprimiendo los **binarios** y sus **dependencias**.

```
$ cd /trabajo/aoetools
$ ar p aoetools_30-3_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
$ ar p libc6_2.12.1-0ubuntu10.2_i386.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
$ ar p lsb-base_4.0-0ubuntu8_all.deb data.tar.gz | tar -zx -C
/tmp/trabajo/descomprimidos
```

Paso 3. Copiando las aplicaciones para el sistema.

```
$ cp -r /tmp/trabajo/descomprimidos/* /
```

Paso 4. Activando el módulo AoE.

```
# modprobe aoe
```

Y ya se encuentra listo para usar.

Todo lo explicado hasta el momento en los pasos anteriores se refiere al trabajo de forma manual en el sistema, es decir, utilizando una terminal y a base de líneas de comando se ejecuta todo el proceso. Para lograr mayor facilidad para utilizar las herramientas aoetools y vblade se diseñó una sencilla interfaz web para cada una en el cliente de administración que ofrece Openfiler. Para esto fue necesario vincular estas herramientas con el cliente de administración lo que supuso la modificación de algunos de los ficheros contenidos en el sistema y la creación de otros. En este proceso se utilizaron los lenguajes PHP, HTML y JavaScript principalmente.

3.2 Interfaz de administración de las nuevas funcionalidades.

Para el mejor entendimiento se especifica que todo el contenido de los servicios y archivos de configuración de Openfiler se encuentran en la dirección `/opt/openfiler/` y el contenido del sitio web de administración se encuentra en `/opt/openfiler/var/www/htdocs/`.

El trabajo realizado se describe a continuación:

Primeramente el usuario tiene que estar autenticado para poder exportar algún disco duro o partición. Se accede a la interfaz de administración de Openfiler con un navegador web en la dirección `https://IP_del_sistema:446`.



Ilustración 9: Ventana de autenticación.

Y una vez logeados se visita la sección **Services** (ver Ilustración 10).

El primer cambio en la interfaz es agregar la opción **Activar/Desactivar (Enable/Disable)** el módulo AoE. Para ello fue necesario editar el fichero

`/opt/openfiler/var/www/htdocs/admin/services.html` con el código correspondiente en PHP para dicha acción.

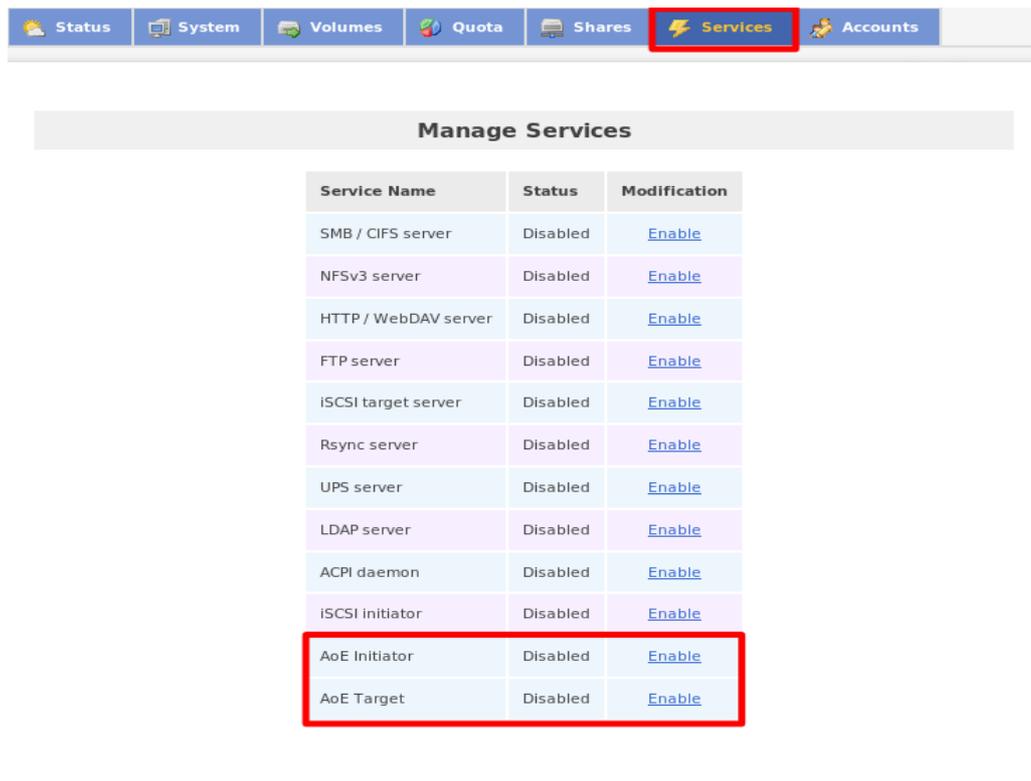


Ilustración 10: Sección Services.

Luego se accede a la sección **Volumes** (ver Ilustración 11).



Ilustración 11: Sección Volumes.

En el panel lateral derecho se muestra una serie de vínculos, se accede a **Block Devices** donde se encuentra una tabla llamada “Block Device Management” que ofrece información acerca de todos los bloques que posee el sistema (ver Ilustración 12). El contenido de la página que se muestra en esa sección se encuentra en el fichero `volumes_physical.html` que está ubicado en la ruta `/opt/openfiler/var/www/htdocs/admin/` del sistema.

Este fichero se modificó, agregando una nueva tabla donde se puede ejecutar otras acciones (ver Ilustración 12) por ejemplo para exportar el volumen al que hace referencia en esa fila de la tabla.

The screenshot shows the Openfiler web interface. On the left, there are two tables. The top table, 'Block Device Management', has columns: Edit Disk, Type, Description, Size, Label type, and Partitions. The bottom table, 'Volume Functions', has columns: Edit Disk, Size, Label type, No. Partitions, Export Disk, Check Disk, and Energy Management. On the right, a sidebar titled 'Volumes section' contains several menu items: Manage Volumes, Volume Groups, Block Devices (highlighted with a red box), Add Volume, iSCSI Targets, and Software RAID. Below that is a 'Support resources' section with links for Report bug, Get support, Forums, and Admin Guide.

| Edit Disk | Type | Description | Size | Label type | Partitions |
|--------------------------|------|---------------|---------|------------|----------------------------|
| /dev/hda | IDE | VBOX HARDDISK | 6.20 GB | msdos | 3 (view) |

| Edit Disk | Size | Label type | No. Partitions | Export Disk | Check Disk | Energy Management |
|--------------------------|---------|------------|----------------|--------------------------|-------------------------|--------------------------|
| /dev/hda | 6.20 GB | msdos | 3 | (export) | (check) | (energy) |

Ilustración 12: Volume Functions.

Ilustración 13: Sección Block Devices.

En el panel lateral derecho se muestra una serie de vínculos, se accede a **Block Devices** (ver Ilustración 13) donde se encuentra una tabla que ofrece información acerca de todos los bloques que posee el sistema. El contenido de la página que se muestra en esa sección se encuentra en el fichero `volumes_physical.html` que está ubicado en la ruta `/opt/openfiler/var/www/htdocs/admin/` del sistema.

Este fichero se modificó, agregando una nueva tabla **Volume Functions** y en ella las columnas **Export Disk**, **Check Disk** y **Energy Management** correspondientes a cada una de las funciones agregadas al sistema.

3.2.1 Interfaz Exportar Volumen.

Para acceder a la interfaz Exportar Volumen se consigue mediante el vínculo **export** en la columna **Export Disk** de la tabla **Volume Functions** (ver Ilustración 14).

| Volume Functions | | | | | | |
|------------------|---------|------------|----------------|----------------------------|---------------------------|----------------------------|
| Edit Disk | Size | Label type | No. Partitions | Export Disk | Check Disk | Energy Management |
| /dev/hda | 6.20 GB | msdos | 3 | (export) | (check) | (energy) |

Ilustración 14: Vínculo Exportar Volumen.

Como se observa en la imagen siguiente el diseño cuenta con 4 columnas, un campo de selección, 3 campos de texto y adjuntos 2 botones, donde:

| Interfaz Exportar Volumen | | | |
|---|----------------------|----------------------|----------------------|
| Dispositivo | Gabinete | Posición | Interfaz de Red |
| <--Seleccione--> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| <input type="button" value="Exportar"/> <input type="button" value="Cancelar"/> | | | |

Ilustración 15: Interfaz Exportar Volumen.

Dispositivo: es el campo donde se selecciona el bloque que se desee exportar.

Gabinete: es el gabinete.

Posición: es la posición en el gabinete que ocupará el bloque.

Interfaz de red: es la interfaz de red que utilizará.

Botón **Exportar:** realiza la acción de exportar una vez entrados correctamente los datos.

Botón **Cancelar:** limpia los campos de entrada de datos, evitando que se realice la acción.

Para el contenido de la página correspondiente a la figura anterior se crea el fichero **exportar_bloques.html** que se encuentra en la dirección del sistema **/opt/openfiler/var/www/htdocs/admin/** el cual contiene el código PHP y HTML que vincula los procesos del sistema necesarios para realizar la acción que se comenta.

3.2.2 Interfaz para Chequeo de Discos.

Para acceder a la interfaz de Chequeo de Discos se consigue mediante el vínculo **check** en la columna **Check Disk** de la tabla **Volume Functions** (ver Ilustración 16).

| Volume Functions | | | | | | |
|------------------|---------|------------|----------------|----------------------------|---------------------------|----------------------------|
| Edit Disk | Size | Label type | No. Partitions | Export Disk | Check Disk | Energy Management |
| /dev/hda | 6.20 GB | msdos | 3 | (export) | (check) | (energy) |

Ilustración 16: Vínculo Chequeo de Disco.

En la página `/opt/openfiler/var/www/htdocs/admin/chequeo_discos.html` del sistema se ofrece la funcionalidad de chequear los diferentes discos del sistema. Esta interfaz se dividió en dos secciones **Vista Estándar** y **Vista Experta** (ver Ilustración 17 y 18 respectivamente).

Vista Estándar
Vista Experta

Seleccione **Vista Experta** solo si conoce **exactamente lo que hace**.

Dispositivo

Chequeo del Disco: /dev/hda

| | | |
|--------------------------------|-----------|----------|
| Partición: | Desmontar | Montar |
| Comprobar sistema de archivos: | Comprobar | Cancelar |

Ilustración 17: Interfaz de Chequeo de Discos, Vista Estándar.

Vista Estándar
Vista Experta

Seleccione **Vista Experta** solo si conoce **exactamente lo que hace**.

Dispositivo

Chequeo del Disco: /dev/hda

| | | |
|--------------------------------|-----------|---------------------------|
| Partición: | Desmontar | Montar |
| Comprobar sistema de archivos: | Comprobar | Cancelar |
| Formatear volumen: | Ejecutar | Ext2 <input type="text"/> |
| Autocomprobación: | Ejecutar | Cancelar |

Ilustración 18: Interfaz de Chequeo de Discos, Vista Experta.

La **Vista Estándar** ofrece las funcionalidades de **Montar** y **Desmontar** la partición que se selecciona previamente en el campo **Dispositivo** perteneciente al disco a que hace referencia la fila en la tabla **Volume Functions**. Y además la funcionalidad de **Comprobar** y **Cancelar** el sistema de archivos de esa misma partición. La **Vista Experta** básicamente ofrece las mismas funcionalidades sin embargo agrega otras dos acciones que por las consecuencias y el significado de ejecutarlas en el sistema se decidieron separar, porque pudieran provocar pérdida de datos con una mala manipulación. Estas acciones son, **Formatear volumen** y **Autocomprobación** de datos S.M.A.R.T.¹⁰ en el disco.

3.2.3 Interfaz Gestión de Energía.

Para acceder a la interfaz de Gestión de Energía se consigue mediante el vínculo **energy** en la columna **Energy Management** de la tabla **Volume Functions** (ver [Ilustración 19](#)).

| Volume Functions | | | | | | |
|------------------|---------|------------|----------------|--------------------------|-------------------------|--------------------------|
| Edit Disk | Size | Label type | No. Partitions | Export Disk | Check Disk | Energy Management |
| /dev/hda | 6.20 GB | msdos | 3 | (export) | (check) | (energy) |

Ilustración 19: Vínculo, Gestión de Energía.

El fichero con el código correspondiente a la interfaz **Gestión de Energía** se encuentra en la dirección del sistema `/opt/openfiler/www/htdocs/admin/gestion_energia.html`.

Gestión de Energía en: /dev/hda

Aplicar cambios

Permitir que el equipo apague el dispositivo para ahorrar energía.

Permitir que este dispositivo reactive el equipo.

[Retornar a la página Block Device Management.](#)

Ilustración 20: Interfaz de Gestión de Energía.

La imagen anterior refleja el contenido de la interfaz visual diseñada para esta funcionalidad.

¹⁰ (Self Monitoring Analysis and Reporting Technology (en español: Tecnología de Auto-Monitorización Análisis e Informe).

Contiene tres casillas de verificación, la primera **Aplicar cambios** verifica que una vez seleccionada una de las opciones de abajo, se pueda ejecutar la acción correspondiente. De no seleccionarse esta casilla no se ejecutará ninguna acción y en consecuencia no habrá ningún cambio en el sistema. Adjuntos aparecen los botones **Ejecutar** para realizar la acción y **Cancelar** para desmarcar todas las opciones que hayan sido seleccionadas en caso de algún arrepentimiento por parte del usuario.

3.3 Pruebas al Sistema.

Para comprobar el correcto funcionamiento del sistema con las nuevas características incorporadas se aplicaron pruebas funcionales a través del método de **caja negra**¹¹, donde se definieron escenarios (ambiente real donde funciona el sistema y arroja resultados). A continuación se expresan los resultados de estas pruebas.

Para la historia de usuario número 1 **Exportar volúmenes por AoE** se definió el escenario siguiente:

| Escenario 1: Exportar Volumen | |
|---|--|
| Descripción: El usuario debe escoger un volumen o partición para exportarlo a través del protocolo AoE. (ver Ilustración 15) | Resultados esperados: - El sistema exporta el volumen seleccionado para ser utilizado por las computadoras clientes. |
| Resultados obtenidos: Se exportó correctamente el volumen. | |
| Observaciones: | |

Tabla 11: Escenario 1: Exportar Volumen.

A continuación se exponen los escenarios correspondientes a la historia de usuario número 2 **Chequear discos duros y particiones.**

| Escenario 2: Chequear Volúmenes (Vista Estándar) | |
|---|--|
| Descripción: El usuario debe escoger un volumen o partición para chequear su correcto funcionamiento. | Resultados esperados: - El sistema comprueba el sistema de ficheros del volumen o partición seleccionada en busca de posibles errores. |
| Resultados obtenidos: Chequeo de volumen correcto. | |
| Observaciones : El sistema solo chequea el volumen seleccionado sólo si soporta comprobación y | |

¹¹ Son aquellas que se enfocan directamente en el exterior del sistema, sin importar el código.

si no está montado, en caso contrario no se efectúa la acción.

Tabla 12: Escenario 2: Chequear Volumen (Vista Estándar).

| Escenario 3: Chequear Volúmenes (Vista Experto) | |
|--|--|
| Descripción: El usuario dispone de las mismas funcionalidades de la Vista Estándar. Además de formatear el volumen escogido y hacer la autocomprobación del sistema. | Resultados esperados: - El sistema formatea el volumen escogido. - El sistema hace la autocomprobación del sistema. |
| Resultados obtenidos: - Formato de volumen correcto. - Autocomprobación satisfactoria. | |
| Observaciones : El sistema verifica que el volumen seleccionado no está montado, en caso contrario no permite formatear o autocomprobar el sistema en dependencia de la acción que se ejecute. | |

Tabla 13: Escenario 3: Chequear Volumen (Vista Experto).

Para la historia de usuario número 3 **Controlar energía de discos duros**, se definió el escenario siguiente:

| Escenario 4: Gestión de Energía | |
|---|--|
| Descripción: El usuario escoge la opción de apagado de disco o en caso contrario escoge a reactivar el disco duro. | Resultados esperados: - El sistema pone a Hibernar el disco duro seleccionado con el consiguiente ahorro de energía. |
| Resultados obtenidos: Disco duro apagado correctamente. Disco duro montado correctamente. | |
| Observaciones : | |

Tabla 14: Escenario 4: Gestión de Energía.

Conclusiones.

El capítulo ofrece una descripción detallada de los pasos llevados a cabo en la implementación de funcionalidades nuevas en el sistema Openfiler. Esto constituye una guía de aprendizaje aplicable a estudios y procesos similares, porque la forma de desarrollo utilizada es una vía estándar de conseguir un producto similar al que propone en este trabajo. Gracias al

proceso seguido y la puesta en práctica de las pruebas a la solución se obtuvo un producto de alta calidad y eficiente para administrar redes de almacenamiento.

Conclusiones Generales.

Las redes de almacenamiento SAN surgen como solución al problema de guardar grandes cantidades de información y su administración constituye un reto. Como respuesta, se da un paso de avance personalizando un sistema operativo libre para su administración convirtiendo el proceso en una actividad que requiere de menos esfuerzos.

Se propuso como sistema a personalizar a Openfiler principalmente por ser de código libre, y contener las herramientas principales para administrar SAN.

Al personalizar Openfiler se agregan nuevas funcionalidades que en la práctica incrementan su versatilidad. Contribuyendo a la utilización de tecnologías de bajo costo pero que brindan servicios de alta calidad, además al ahorro de energía eléctrica, recurso de vital importancia en el país.

Se demuestra con la investigación la adaptabilidad del software libre a las necesidades del usuario, y su aprovechamiento en beneficio de la sociedad.

Recomendaciones.

- Hacer un estudio similar con la distribución cubana GNU/Linux Nova para lograr un producto netamente cubano para la administración de redes de almacenamiento.
- Descargar el repositorio de aplicaciones de Openfiler para que la actualización e instalación de los programas en el sistema sea más sencilla.
- Mejorar en Openfiler las funcionalidades agregadas en esta investigación e incorporar nuevas para mejorar sus prestaciones.

Referencias Bibliográficas.

- [1] Mundo Cisco. *Que es una red SAN*. Septiembre 2009. [Disponible en: <http://www.mundocisco.com/2009/09/que-es-una-red-san.html>. [Citado: diciembre de 2010].
- [2] Larousse Planeta S.A. *Gran diccionario de la Lengua Española*, 1996. [Citado: diciembre de 2010].
- [3] TextosCientificos.com. [Disponible en: <http://www.textoscientificos.com/informatica/almacenamiento/cintas-magneticas>] [Citado: diciembre 2010].
- [4] Unidades de discos duros. *IDE*, 2006. [Disponible en: <http://www.harddiskhome.com/spanish/Unidades-de-Discos-de-IDE.html>] [Citado: enero de 2011].
- [5] La interfaz SCSI. *Tecnología del PC*, 1990-2010. [Disponible en: http://www.zator.com/Hardware/H6_3.html] [Citado: enero de 2011].
- [6] Microsoft-TechNet. [Disponible en: <http://technet.microsoft.com/es-es/library/aa995741%28EXCHG.65%29.aspx>] [Citado: enero 2011].
- [7] WordPress.com. *Tecnología de storage compartido*, 21 febrero 2008. [Disponible en: <http://es.wordpress.com/tag/direct-attached-storage/>] [Citado: enero 2011].
- [8] Infoviews México. [Disponible en: <http://www.infoviews.com.mx/LaCie/NAS/#NAS>] [Citado: enero 2011].
- [9] TopBits.com. [Disponible en: <http://www.tech-faq.com/network-attached-storage.html>] [Citado: enero 2011].
- [10] TicoStyle. *SAN, definición, características, ventajas, componentes*. [Disponible en: <http://ticostyle.wordpress.com/>] [Citado: enero 2011].
- [11] NesTux. [Disponible en: <http://nestux.com/blog/red-hat-enterprise-linux-6-disponible>] [Citado: enero 2011].
- [12] Nexenta. *Características de OpenSolaris*. [Disponible en: <http://nexenta.wordpress.com/2008/05/04/caracteristicas-de-opensolaris/>] [Citado: febrero 2011].
- [13] Kernelia. [Disponible en: <http://www.kernelia.com/index.php/soluciones-open-source/openfiler>] [Citado: febrero 2011].
- [14] Openfiler, *Sitio Oficial*. [Disponible en: <http://www.openfiler.com/products/system-requirements>] [Citado: febrero 2011].
- [15] Linux Hard Labs. [Disponible en:

<http://tuxedlinux.wordpress.com/category/servidores/page/3/>] [Citado: febrero 2011].

[16] S. Saether Bakken, A. Aulbach, E. Schmid, Jim Winstead, Lars Torben Wilson, R. Lerdorf, Zeev Suraski, Andrei Zmievski, y Jouni Ahto. *PHP Manual*, 2001. [Citado: febrero 2011].

[17] C. Ramey, B. Fox: "*Bash Reference Manual, v2.5a*". Free Software Foundation, 2001. [Citado: febrero 2011].

[18] Geany. *Sitio Oficial*, 2005-2010. [Disponible en: <http://www.geany.org/Main/About>] [Citado: febrero 2011].

[19] Mas, Jordi. Megías Jiménez, David. Gibert Ginestà, Marc. Peña González, Álvaro. *Ingeniería del software en entornos de Software Libre*. Marzo 2005. [Citado: marzo 2011].

[20] Poppendieck M., Poppendieck T. "*Lean Software Development: An Agile Toolkit for Software Development Managers*". Addison Wesley, 2003. [Citado: marzo 2011].

Bibliografía.

1. Álvarez de Zayas, C. *METODOLOGÍA DE LA INVESTIGACIÓN CIENTÍFICA*. Santiago de Cuba, Universidad de Oriente, 1995. [Consultado: noviembre 2010].
2. Blanco Pérez, A., Hernández F. *Acerca de la orientación y asesoría del trabajo científico de maestrantes y doctorantes*. [Consultado: diciembre 2010].
3. Coraid. *Product*. [Disponible en: <http://www.coraid.com/products>] [Visitado: diciembre 2010].
4. DesarrolloWeb.com. [Disponible en <http://www.desarrolloweb.com>] [Visitado: marzo 2011].
5. Guardo García, M. E. *Los Componentes del Diseño Teórico de la Investigación Científica. Una Reflexión Praxiológica. Components of the Theoretical Design of the Scientific Investigation. A Praxiological Reflection*. Revista Pedagogía Universitaria, Vol. XIV No. 3. Matanzas, 2009. [Consultado: enero 2011].
6. Hernández E. *Cómo escribir una tesis*. Escuela Nacional de Salud Pública, 2006. [Consultado: diciembre 2010].
7. IBM. Estructura SAN. *Switch de estructura simples y accesibles, y Enterprise SAN Directors de alta disponibilidad*. [Disponible en: http://www.ibm.com/co/systems/storage/san/sanfabric_ent.phtml] [Visitado: diciembre 2010].
8. Hard Disk Sentinel. *Can we believe S.M.A.R.T. ?* [Disponible en: <http://www.hdsentinel.com/smart/>] [Visitado: febrero 2011].
9. Linux Magazine. *Software RAID on Linux with mdadm*. [Disponible en: <http://www.linux-mag.com/id/7939/>] [Visitado: marzo 2011].
10. Badal, M. *Elaboración de referencias y citas según las normas de la American Psychological Association (APA)*, 5ta Edición. [Consultado: enero 2011].
11. Microsoft TechNet. *Introducción a Administración de discos*. [Disponible en: <http://technet.microsoft.com/es-es/library/cc737216%28WS.10%29.aspx>] [Visitado: enero 2011].
12. Netmedia.Info. *Supera cantidad de información capacidad para almacenarla*. [Disponible en: <http://www.netmedia.info/soluciones/software/supera-cantidad-de-informacion-capacidad-para-almacenarla/>] [Visitado: abril 2011].
13. PhpClasses.org. [Disponible en <http://www.phpclasses.org>] [Visitado: marzo 2011].
14. Repositorio de Ubuntu. [Disponible en: <http://cu.archive.ubuntu.com/ubuntu/>] [Visitado: febrero 2010].
15. Source Force. *Smartmontools Download and Installation*. [Disponible en: <http://sourceforge.net/apps/trac/smartmontools/wiki/Download>] [Visitado: abril 12, 2011].
16. Hard Disk Sentinel. *SMART. How Hard Disk S.M.A.R.T. Works?* [Disponible en:

<http://www.hdsentinel.com/smart/index.php>] [Visitado: abril 2011].

17. Hard Disk Sentinel. *S.M.A.R.T. attribute list (ATA)*. [Disponible en: <http://www.hdsentinel.com/smart/smartattr.php>] [Visitado: abril 2011].

Anexos.

Anexo 1. Entrevista realizada a integrante de la empresa cubana GEDEME.

Entrevista sobre las herramientas y los sistemas de administración de redes de almacenamiento.

Autor/es: Adrian Martínez Casero y Yaniel Arzuaga Ruíz.

El propósito de esta entrevista es determinar las herramientas y sistemas de administración de redes de almacenamiento más utilizados. Para ello esta encuesta esta dirigida al compañero Orestes Chavez de la empresa GEDEME, pues ellos tienen gran experiencia en la confección de medios informáticos y electrónicos.

Preguntas:

¿Qué es una red de almacenamiento?

¿Qué importancia tiene la administración de las redes de almacenamiento?

¿Preferiría un sistema de código abierto o un sistema privativo para administrar una red de almacenamiento? ¿Por qué?

¿Qué herramientas son las que más se utilizan en la administración de redes de almacenamiento?

¿Conoce de la existencia de alguna otra herramienta con este fin?

¿Cuáles sistemas conoce que se utilizan más en la administración de redes de almacenamiento?

¿Cree que es más factible utilizar un sistema privativo o un sistema de código abierto? ¿Por qué?

¿Qué herramientas cree que no deberían faltar en un sistema de administración de redes de almacenamiento?

Anexo 2. Listado total de los paquetes instalados y sus dependencias.

| Paquete | Disponible en: |
|----------------|---|
| aoetools | http://cu.archive.ubuntu.com/ubuntu/pool/main/a/aoetools/aoetools_23-0ubuntu5_i386.deb |

| | |
|----------------|---|
| libc6 | |
| lsb-base | http://cu.archive.ubuntu.com/ubuntu/pool/main/l/lsb/lsb-base_3.1-5ubuntu2_all.deb |
| vblade | http://cu.archive.ubuntu.com/ubuntu/pool/main/v/vblade/vblade_20-1ubuntu1_i386.deb |
| vblade-persist | |
| runit | |
| dosfstools | http://cu.archive.ubuntu.com/ubuntu/pool/main/d/dosfstools/dosfstools_2.11-2.1ubuntu1_i386.deb |

Tabla 15: Listado de Paquetes.

Anexo 3. Documentación sugerida para las aplicaciones vblade y aoetools.

Documentación sugerida para las aplicaciones vblade y aoetools dentro del sistema.

/usr/share/doc/aoe6-59/aoetools-23/COPYING

/usr/share/doc/aoe6-59/aoetools-23/HACKING

/usr/share/doc/aoe6-59/aoetools-23/NEWS

/usr/share/doc/aoe6-59/aoetools-23/README

/usr/share/doc/aoe6-59/aoetools-23/TODO

/usr/share/man/man8/aoe-discover.8.gz

/usr/share/man/man8/aoe-flush.8.gz

/usr/share/man/man8/aoe-interfaces.8.gz

/usr/share/man/man8/aoe-mkdevs.8.gz

/usr/share/man/man8/aoe-mkshelf.8.gz

/usr/share/man/man8/aoe-revalidate.8.gz

/usr/share/man/man8/aoe-stat.8.gz

/usr/share/man/man8/aoe-version.8.gz

/usr/share/man/man8/aoecfg.8.gz

/usr/share/man/man8/aoeping.8.gz

Anexo 4. Diagramas.

Diagrama de dominio.

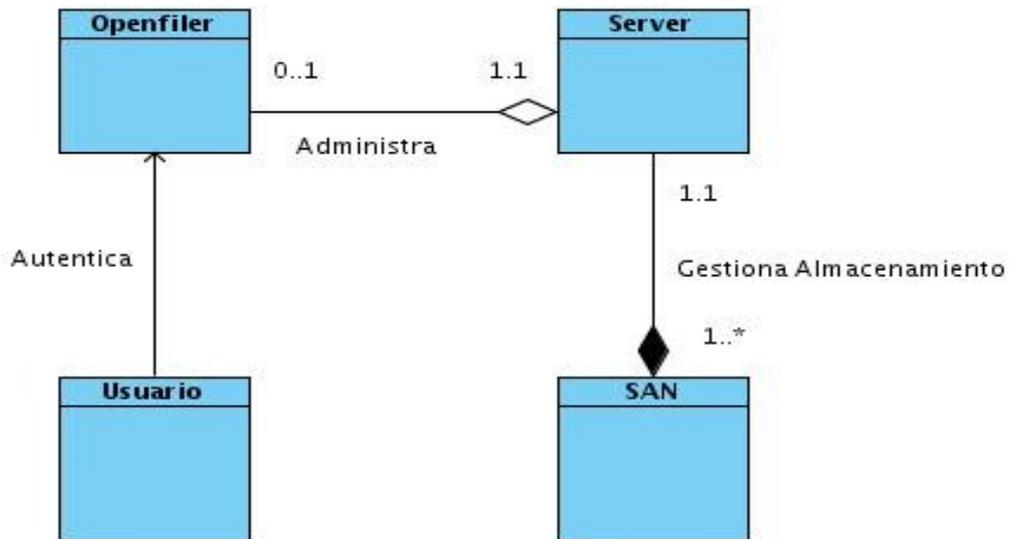


Ilustración 21: Diagrama de Dominio.

Diagrama de Despliegue.

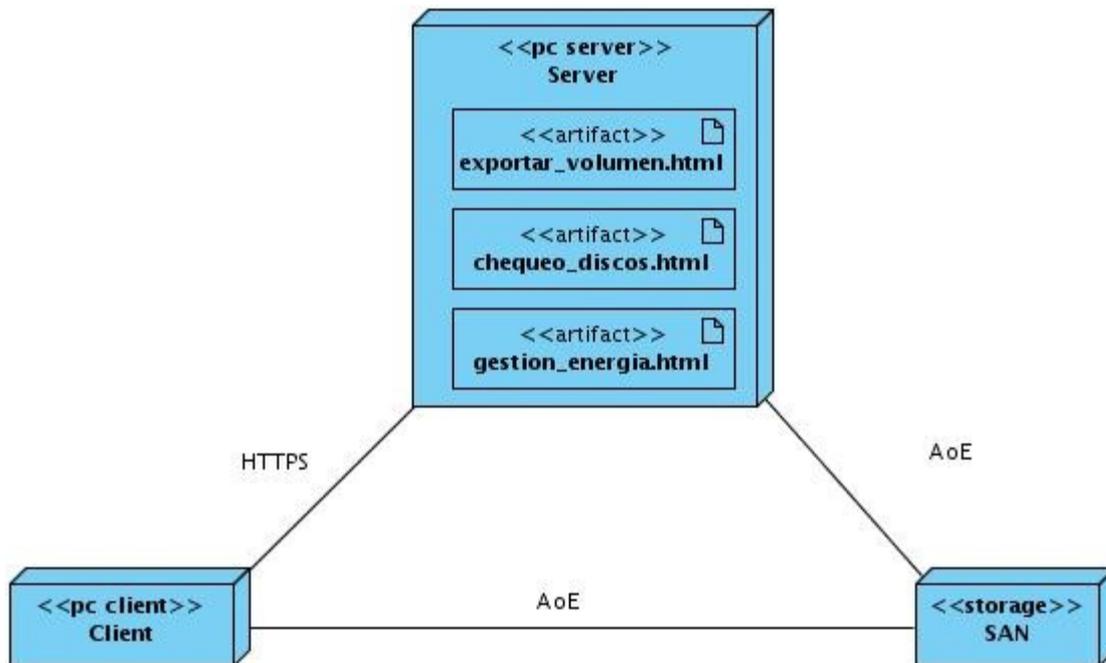


Ilustración 22: Diagrama de Despliegue.

Anexo 5. Fragmentos de código.

```
<h3 align="center">Interfaz Exportar Volumen</h3>
<form action="" method="post">
  <div align="center">
    <table cellpadding="2" cellspacing="8" border="0">
      <tr bgcolor="" <?php print(($GLOBALS["color_table_heading"])); ?> ">
        <td><strong>Dispositivo</strong></td>
        <td><strong>Gabinete</strong></td>
        <td><strong>Posición</strong></td>
        <td><strong>Interfaz de Red</strong></td>
      </tr>
      <tr>
        <td class="color_table_row1" bgcolor="">
          <select name="file_disk" title="Seleccione un dispositivo.">
            <option value="defecto"><--Seleccione--></option>
            <?php
              for($i = 0; $i < count($of_disks); $i++)
              {
                for ($j = 0; $j < count($of_disks[$i]["partitions"]); $j++)
                {
                  print("<option>" . $of_disks[$i]["partitions"][$j]["device"] . "</option>");
                }
              }
            <?>
          </select>
        </td>
        <td class="color_table_row1" bgcolor=""><input type="text" size="8" name="shelf" title="Gabinete (Ej">
        <td class="color_table_row1" bgcolor=""><input type="text" size="8" name="slot" title="Posición en el">
        <td class="color_table_row1" bgcolor=""><input type="text" size="8" name="netif" title="Interfaz de r">
      </tr>
    </table>
  </div>
  <input type="submit" name="export_volume" value="Exportar" title="Clic para exportar el dispositivo." onclick=">
  <input type="reset" value="Cancelar" title="Clic para limpiar los campos." />
</form>
```

Ilustración 23: Implementación de la interfaz web Exportar Volúmenes.

```
if(($shelf!="") && ($slot!="") && ($netif!="") && ($file_disk!="defecto") && (!$error_disco) && (!$error_g
{
  exec("/usr/bin/sudo /usr/sbin/vblade " . $shelf . " " . $slot . " " . $netif . " " . $file_disk);|
  print("<br>");
  print("<br>");
  print("<br>");
  print("<strong><font color='green'>Ha exportado el dispositivo </font>". $file_disk ." <font color="
  print("Retornar a la sección <a href=volumes_physical.html>Block Device Management.</a><br />");
}
}
```

Ilustración 24: Implementación de la funcionalidad Exportar Volúmenes.

```
if(isset($_POST['cmp_fs'])) //boton para comprobar sistema de ficheros
{
  exec("/usr/bin/sudo /sbin/fsck " . $file_disk." -cfy");
}

if(isset($_POST['cancel_cmp_fs'])) //boton para cancelar comprobacion fs
{
  exec("/usr/bin/sudo /usr/bin/killall /sbin/fsck " );|
}
}
```

Ilustración 25: Implementación de la funcionalidad Chequeo de Discos.

```

if(isset($_POST["enviar"]))
{
    if(isset($primero))
    {
        for($i=0; $i < count($of_disks); $i++)
        {
            exec("/usr/bin/sudo /bin/umount ". $of_disks[$i]["device"] );
            break;
        }
        for($i=0; $i< count($of_disks); $i++)
        {
            exec("/usr/bin/sudo /sbin/hdparm -y ". $of_disks[$i]["device"] );
            break;
        }
    }

    if(isset($segundo))
    {
        for($i=0; $i< count($of_disks); $i++)
        {
            exec("/usr/bin/sudo /bin/mount ". $of_disks[$i]["device"] ." /mnt" );
            break;
        }
    }
}

```

Ilustración 26: Implementación de la funcionalidad para controlar energía de discos.