

**Centro de Software Libre  
Facultad 1**



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

**Título:** *Sistema de archivos distribuidos en Nova para  
Servidores.*

**Autora:** *Iracema Coset Alvisua Velazquez*

**Tutores:** Ing. Abel Alfonso Fírvida Donéstevez

**La Habana. 18 de junio de 2012  
"Año 54 de la Revolución"**

## DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del trabajo titulado “Sistema de archivos distribuidos en Nova para Servidores” y autorizo al Centro de Software Libre de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

**Iracema Coset Alvisua Velazquez**

\_\_\_\_\_

Firma del Autor

**Ing. Abel Alfonso Fírvida Donéstevez**

\_\_\_\_\_

Firma del Tutor

## AGRADECIMIENTOS

*A mi mamá por ser mi motor impulsor, por apoyarme en todo, por preocuparse por mí y haber estado siempre que la he necesitado, por ser la mejor madre de este mundo. Te quiero mucho.*

*A mi papá por haberme dado la luz de mi carrera, por preocuparse por mis resultados docentes.*

*A mi padrastro Ony por comportarse conmigo como si fuera mi padre, por ser tan bueno con mi mamá.*

*A Surella por ser mi tía-abuela y mi profesora de matemáticas, por quererme y preocuparse por mí como a su hija.*

*A mi abuela Lolita que aunque ya no este hubiese disfrutado mi triunfo.*

*A mis tíos Walter, Kiry y Ely por sus profundos consejos.*

*A mi abuela María por ser como es ella conmigo.*

*A Yoani, Eiger y Albin por ayudarme con mucha dedicación cuando lo he necesitado.*

*A Miguel Ángel por ser la persona que abrió la puerta de mi camino, porque sin su dedicación, preocupación y comprensión en los tres primeros años de mi carrera no hubiese podido llegar tan lejos.*

*A Yoanni Guerra por ser un amigo incondicional, por tenerme siempre presente y haberme ayudado en estos largos cursos de mi carrera.*

*A mi compañero de proyecto Eugenio por su locura.*

*A Yisy, Yisel, Yanela, Yeney, Yaisy, Mirley y Susy por haber sido mi familia en la universidad, porque cuando la soledad me acompañaba ustedes estuvieron ahí para escucharme, para aconsejarme y por perdonarme todas mis malcriadeces y ayudarme cuando lo he necesitado.*

*A mi novio Miguel Enrique por ayudarme en lo más mínimo, por darme su cariño, por tener el poder de soportarme, por hacerme fuerte, por demostrarme que soy importante para él, por elegirme a mí, por su forma de expresar lo que siente.*

*A Julio, Harol y Jorgito por haberme dado un curso lleno de risas y de bienestar.*

*Especialmente a mi tutor por ser muy dedicado, preocupado y tener siempre una palabra de aliento cuando me he sentido por el piso, por no tener en cuenta los sábados y domingos para reunirnos, por ser la mejor persona de esta universidad.*

DEDICATORIA

*A mi mamá por ser la mejor madre de este mundo.*

*A mi papá por permitirme ser su divino tesoro.*

## RESUMEN

De gran utilidad es el hecho de poder compartir la información de forma tal que esté disponible desde todas las estaciones de trabajo de una empresa, pues resulta factible disponer de algún mecanismo que posibilite el almacenamiento de forma global. Una solución inteligente es la utilización de los sistemas de archivos distribuidos. Nova para servidores en su versión más reciente incluye a Zentyal que es una solución de servidor fácil de administrar basada en Ubuntu. Con la realización de este trabajo se desarrolló un módulo para la plataforma Zentyal, con el objetivo de que pueda brindar el servicio de sistema de archivos distribuidos. Se realizó un estudio de los sistemas de archivos distribuidos existentes a nivel mundial, caracterizando principalmente tres de ellos, que son los más utilizados. Se utilizó la metodología de desarrollo de software OpenUp. Se analizaron y diseñaron los Casos de Uso del sistema luego se implementaron y se le hicieron pruebas de caja negra a cada uno, constatando su correcto funcionamiento.

**Palabras clave:** servidores, Sistema de archivos distribuidos, Zentyal

## Tabla de contenido

Introducción .....	1
Capítulo 1.- Fundamentación teórica. ....	3
1.1.    Introducción.....	3
1.2.    Conceptos fundamentales. ....	3
1.3.    Estudio de los Sistemas de archivos distribuidos.....	3
1.3.1.    CODA.....	4
1.3.2.    OpenGFS .....	8
1.3.3.    GlusterFS.....	13
1.4.    Sistema de archivos distribuidos para la solución. ....	21
1.5.    Marco de desarrollo.....	23
1.6.    Conclusiones del capítulo.....	26
Capítulo 2.- Análisis y diseño del sistema. ....	27
2.1.    Introducción .....	27
2.2.    Actores del sistema .....	27
2.3.    Requisitos funcionales .....	28
2.4.    Requisitos no funcionales .....	30
2.5.    Diagrama de caso de uso del sistema .....	32
2.6.    Descripción de los Casos de uso del sistema .....	34
2.6.1.    Caso de uso Gestionar Servidores .....	34
2.6.2.    Caso de uso Insertar Volumen .....	36
2.6.3.    Caso de uso Insertar Bloque .....	37
2.6.4.    Caso de uso Modificar Volumen .....	38
2.6.5.    Caso de uso Modificar Bloque .....	39
2.6.6.    Caso de uso Mostrar Volumen.....	40
2.6.7.    Caso de uso Mostrar Bloques .....	40
2.6.8.    Caso de uso Eliminar Volumen .....	41
2.6.9.    Caso de uso Eliminar Bloque.....	42

2.6.10.	Caso de uso Administrar servicio de Volumen .....	42
2.6.11.	Caso de uso Administrar Servicio General .....	44
2.6.12.	Caso de uso Administrar Plugin .....	45
2.6.13.	Caso de uso Administrar Firewall.....	47
2.6.14.	Caso de uso Administrar Firewall.....	47
2.7.	Diagrama de clases del diseño .....	48
2.8.	Diagrama de clases del diseño con estereotipos web .....	50
2.9.	Diagrama de interacción .....	51
2.9.1.	Diagrama de secuencia. ....	51
2.10.	Arquitectura de software.....	53
2.10.1.	Modelo Vista Controlador.....	53
2.11.	Patrones de diseño .....	54
2.11.1.	Patrones GRASP .....	54
2.11.2.	Experto en información .....	54
2.11.3.	Creador .....	54
2.11.4.	Alta Cohesión .....	55
2.11.5.	Bajo Acoplamiento.....	55
2.12.	Diagrama de despliegue.....	55
2.13.	Diagrama de componente .....	56
2.14.	Conclusiones del capítulo .....	58
Capítulo 3.	Implementación y prueba del sistema. ....	58
3.1.	Introducción .....	58
3.2.	Pruebas de caja negra .....	58
3.3.	Conclusiones del capítulo.....	64
Conclusiones	.....	65
Recomendaciones	.....	66
Referencias bibliográficas	.....	67
Bibliografía.....	.....	68

Glosario de términos.....	69
---------------------------	----

## **Índice de figuras**

Figura 1 Representación de los resultados arrojados.....	23
Figura 2 Diagrama de casos de uso del sistema.....	34
Figura 3 Diagrama de clases del sistema.....	49
Figura 4 Diagrama de clases con estereotipos web.....	51
Figura 5 Diagrama de secuencia de CU del Gestionar bloque.....	53
Figura 6 Diagrama de despliegue del sistema.....	56
Figura 7 Diagrama de componentes del sistema.....	57

## **Índice de tablas**

Tabla 1 Propiedades de GlusterFS.....	16
Tabla 2 Características relevantes de los sistemas de archivos distribuidos.....	21
Tabla 3 Comparación de los sistemas de archivos distribuidos.....	22
Tabla 4 Actores del sistema.....	27
Tabla 5 Descripción del CU Gestionar servidores.....	34
Tabla 6 Descripción del CU Insertar volumen.....	36
Tabla 7 Descripción del CU Insertar bloque.....	37
Tabla 8 Descripción del CU Modificar volumen.....	38
Tabla 9 Descripción del CU Modificar bloque.....	39
Tabla 10 Descripción del CU Mostrar volúmenes.....	40
Tabla 11 Descripción del CU Mostrar bloques.....	40
Tabla 12 Descripción del CU Eliminar bloque.....	42
Tabla 13 Descripción del CU Administrar servicio de volumen.....	42

Tabla 14 Descripción del CU Administrar servicio general.....	44
Tabla 15 Matriz parcial del escenario para el Caso de uso Gestionar servidores.....	59
Tabla 16 Matriz de caso de prueba para el Caso de uso Gestionar servidores.....	59
Tabla 17 Matriz parcial del escenario para el Caso de uso insertar volumen.....	60
Tabla 18 Matriz de caso de prueba para el Caso de uso Insertar volumen.....	60
Tabla 19 Matriz parcial del escenario para el Caso de uso Insertar bloque.....	62
Tabla 20 Matriz de caso de prueba para el Caso de uso Insertar bloque.....	62
Tabla 21 Matriz parcial del escenario para el Caso de uso Modificar volumen.....	63
Tabla 22 Matriz de caso de prueba para el Caso de uso Modificar volumen.....	63
Tabla 23 Matriz parcial del escenario para el Caso de uso Modificar bloque.....	64
Tabla 24 Matriz de caso de prueba para el Caso de uso Modificar bloque.....	64

## Introducción

De gran utilidad es el hecho de poder compartir la información de forma tal que esté disponible desde todas las estaciones de trabajo de una empresa, para ello se utilizan comúnmente los sistemas de archivos distribuidos que no son más que el diseño de espacio de usuario amigable sin comprometer el rendimiento, sirviendo para compartir archivos, impresoras y otros recursos como un almacenamiento persistente en una red de computadoras.

Nova para servidores en su versión más reciente incluye a Zentyal que es una plataforma para la administración de servidores. Esta herramienta brinda una serie de servicios al usuario pero no incluye la prestación del servicio de sistemas de archivos distribuidos.

El problema descrito anteriormente denota la siguiente situación problemática: Nova para Servidores cuenta con varias aplicaciones para crear un sistema de archivos distribuidos pero no con una aplicación que permita su implantación de forma intuitiva por parte del administrador.

Se define como problema científico: ¿Cómo garantizar la implantación de un sistema de archivos distribuidos en Nova para Servidores?

Se determina como objeto de estudio: Los sistemas de archivos distribuidos, acotándolo al campo de acción el sistema de archivos distribuidos en Nova. Se identifica como objetivo general de la investigación: Desarrollar un módulo para Zentyal que permita la implantación de un sistema de archivos distribuidos de forma cómoda desde un ordenador.

Se define como objetivos específicos de la investigación, los siguientes:

1. Sistematizar tendencias actuales de los sistemas de archivos distribuidos.
2. Asimilar la implementación de módulos para Zentyal.
3. Desarrollar un módulo de Zentyal para la implantación de un sistema de archivos distribuidos.
4. Validar el buen funcionamiento del desarrollo del módulo de Zentyal para la implantación de un sistema de archivos distribuidos.

La investigación se sustenta en la siguiente idea a defender: El desarrollo de un módulo de Zentyal para la conexión automática del sistema de archivos distribuidos podría eliminar lo engorroso de su realización manualmente.

Para desarrollar el siguiente trabajo se hace necesario la utilización de los siguientes métodos teóricos y empíricos Dentro de los métodos teóricos se trabajará con el de Análisis-Síntesis. Que permitirá estudiar de forma separada los sistemas de archivos distribuidos para así comprender mejor su funcionamiento. Dentro de los métodos empíricos se utilizará el método de observación para constatar el buen funcionamiento de la aplicación.

## **Resultado esperado**

Obtención de una aplicación que posibilite la implantación de un sistema de archivos distribuidos en Nova para servidores desde un ordenador.

El documento de la presente investigación está estructurado en tres capítulos, en cada uno se abordará temas importantes para el desarrollo del mismo.

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

En este capítulo se describe el estudio del estado del arte de los diferentes tipos de sistema de archivos distribuidos así como sus principales características. Se realiza una breve panorámica sobre la metodología a utilizar y el lenguaje de programación. Se explicará además qué es la plataforma Zentyal.

## **Capítulo 2: Análisis y diseño del sistema.**

Abarcará todo el análisis y diseño del módulo que se desea implementar. Para ello se precisa una previa definición de los requerimientos funcionales y no funcionales.

## **Capítulo 3: Implementación y pruebas del sistema.**

Se diseñarán y ejecutarán pruebas de caja negra para comprobar los resultados obtenidos. Se analizarán los resultados obtenidos y se arribará a conclusiones.

## Capítulo 1.- Fundamentación teórica.

### 1.1. Introducción.

En el presente capítulo se describen los conceptos fundamentales de la investigación. Se realiza un estudio del estado del arte, con el objetivo de analizar las características y funcionalidades de un conjunto de sistemas de archivos distribuidos; luego se realiza una comparación entre ellos y finalmente se seleccionan las tecnologías a utilizar durante la solución propuesta.

### 1.2. Conceptos fundamentales.

Varios son los conceptos asociados a los sistemas de archivos distribuidos. A continuación se muestran algunos de ellos:

Según Kelly López: “Los sistemas distribuidos de archivos permiten a los programas almacenar y acceder a archivos remotos del mismo modo que si fuera local, permitiendo a los usuarios que accedan a los archivos desde cualquier computador”. (Kelly López 2012)

Según la Universidad Nueva Esparta:” Un sistema de archivos distribuidos es una variación del modelo de sistema de archivos de tiempo compartido, donde varios usuarios comparten información. Es un sistema de archivos en el que los usuarios, servidores, y dispositivos de almacenamiento están dispersos en una red con una distribución transparente para los usuarios. El propósito del sistema de archivos distribuidos es permitir un igual acceso y comparación cuando los archivos se encuentran físicamente dispersos en varios lugares de un sistema distribuido”. (Universidad Nueva Esparta 2011)

Según la Guía de Administración de GlusterFS: “Un sistema de archivos distribuidos (SAD), es el software encargado de administrar tanto lógica como físicamente los datos contenidos en uno o varios dispositivos secundarios los cuales no se encuentran centralizados, es decir, pueden localizarse en diferentes computadoras interconectadas entre sí mediante una red”, siendo esta última la más acertada para el contexto de esta investigación.

### 1.3. Estudio de los Sistemas de archivos distribuidos

Para la selección de un sistema de archivos distribuidos se hará un estudio de las características de tres de ellos: CODA, GlusterFS, OpenGFS. Finalmente se hará un cuadro comparativo con el propósito seleccionar el más idóneo para la integración del módulo.

La caché en un sistema de archivos distribuidos garantiza la disponibilidad de los datos cuando un servidor de archivos no está disponible. Para que este mecanismo funcione correctamente la información de los servidores se almacena en réplicas. Los sistemas de archivos distribuidos como CODA incluyen su propio mecanismo de gestión de volúmenes que simplifican la gestión del almacenamiento compartido mediante el uso de réplicas.

Por otra parte GlusterFS proporciona servicios de gestión de volúmenes, aprovechando esto para montar volúmenes de diferentes servidores de archivos en una jerarquía de directorios centrales que es soportado por el sistema de ficheros. Estas jerarquías de directorios son visibles para todos los clientes del sistemas de ficheros distribuidos. Permitiendo a los usuarios acceder a sus archivos de datos exactamente de la misma manera que desde cualquier cliente. Si la computadora falla, se puede usar otra pues los archivos aún están intactos y con conexiones de seguridad en el servidor de archivos distribuidos.

Los sistemas de archivos distribuidos son independientes del tipo de computadora de escritorio utilizada como cliente, lo que permite a los usuarios seleccionar la computadora que mejor se adapte a sus necesidades y acceder al sistema de archivos distribuidos de forma centralizada.

### 1.3.1. CODA

El sistema de archivos distribuidos CODA está basado en AFS21. Su arquitectura dispone con tres componentes: clientes, servidores y el controlador del sistema.

Los clientes son ordenadores utilizados por los usuarios para acceder al sistema de archivos distribuidos.

Los servidores constituyen ordenadores con elevado nivel de seguridad y confiabilidad. Su propósito es gestionar el servicio de compartición de archivos solicitados por los clientes. Para establecer una conexión con un servidor es necesario implementar los protocolos de autenticación requeridos.

El controlador del sistema tiene como objetivo propiciar un único punto para el control de administración.

### Conceptos claves

**Celdas:** Una celda es un grupo de servidores que comparten un conjunto de bases de datos de

---

<sup>1</sup> AFS2: (Andrews File System) es un sistema de archivos distribuidos.

configuración común. De entre todos los servidores, uno tiene que ser la máquina que controla el sistema SDM (Sistema de Control de la Máquina). (F.J.Mora 2008)

Encargado de las siguientes funciones:

- Equipo que controla el sistema, entendiendo por sistema a la celda que pertenece el SCM.
- Único equipo que modifica la configuración.
- Propaga la configuración al resto.

**Volumen:** Un volumen es más pequeño que una partición y más grande que un directorio. Los volúmenes tienen una raíz y contiene un árbol de directorio. Cada volumen está montado en algún lugar bajo /coda y forma un árbol de /coda.

**Almacenamiento de datos:** Los servidores Coda almacenan los archivos, identificándolos por un número, en el árbol de directorios /vicepa. Los metadatos (propietarios, listas de control de acceso, vectores de versión) se almacenan en un fichero de datos RVM. (Sergio Gonzalez 2004)

**RVM:** es la abreviatura de Recoverable Virtual Memory (Memoria Virtual Recuperable). Es una librería transaccional que forma parte del espacio de direcciones virtual del proceso persistente del disco y copia los cambios a esta memoria automáticamente desde el almacenamiento persistente. Coda usa RVM para administrar sus metadatos. Estos datos son almacenados en un archivo RVM que es mapeado a la memoria nada más con arrancar. Las modificaciones se hacen en la VM y son escritas en el archivo de log RVM cuando una transacción es realizada. (Sergio Gonzalez 2004)

**Validación:** Cuando Coda detecta que un servidor está disponible de nuevo, este validará los datos de la caché antes de usarlos, para asegurarse que los datos de la caché estén actualizados. El cliente Coda compara las etiquetas de versión de la caché asociadas a cada objeto con las etiquetas de versión del servidor. (Sergio Gonzalez 2004)

## Características

- Altas prestaciones mediante caché persistente en los equipos clientes.
- Replicación de servidores.
- Buena escalabilidad.
- Adaptación al ancho de banda.
- Funcionamiento continuo ante pérdidas de conexión.

- Funcionamiento sin conexión (equipos móviles)
- Modelo de seguridad para autenticación, encriptación y control de acceso.
- Es un árbol de directorios con una raíz.
- Más pequeño que una partición.
- Más grande que un directorio.
- Los clientes montan los volúmenes dentro de su directorio /coda (todos los clientes ven lo mismo)

## Puntos de montaje

Existe un volumen principal que se monta en /coda. Se puede ampliar el árbol insertando otros volúmenes. Para ello se utiliza el siguiente comando: **cfs mkmount <nv><pa>**

**nv:** nombre del volumen

**pa:** punto de anclaje

Los puntos de anclaje son persistentes (no necesitan ser renombrados al reiniciar)

## Caché en el cliente

- Caché persistente ubicada en /usr/coda/venus.cache.
- El cliente también tiene metadatos en un RVM (/usr/coda/DATA)
- La caché es una réplica de los archivos del servidor.
- Los accesos de lectura son rápidos ya que son locales (usa caché).
- Permite el acceso en desconexión. CODA sincroniza la caché de los clientes con los archivos del servidor.

## Autenticación

Hace uso de la autenticación de testigos para comprobar el acceso de los usuarios a los archivos. Los usuarios se autentican con el que obtienen una clave de entrada.

## Soporte en el cliente

CODA necesita un soporte mínimo en el kernel (módulo). Trabaja en unión con el administrador de la caché (Venus). Venus es la aplicación que trabaja en el espacio de usuario encargado de montar la información que es replicada dentro de /coda.

## Utilidades

- **cfs:** Listas de control de acceso, cache, puntos de anclaje.
- **codacon:** Realiza la monitorización de operaciones de administración de la caché.
- **clog:** Permite la identificación en el servidor.
- **cmon:** Muestra la lista de servidores.
- **ctokens:** Muestra una lista de tokens CODA de los usuarios que se autenticaron.

## Servicios que se ejecutan en el servidor

Los servicios que se ejecutan en el servidor son los siguientes:

- **codasrv:** Servidor principal, que trabaja en conjunto con el proceso venus en los clientes. Estos procesos permiten el intercambio de datos entre cada una de las máquinas que conforman el cluster.
- **auth2:** Acepta peticiones de clog<sup>2</sup> y gestiona los testigos. Se ejecuta en todos los servidores; su función es la de validar a los usuarios. Las contraseñas solo pueden ser modificadas en el servidor maestro (SCM), por lo cual la copia de la base de datos tiene la propiedad de solo lectura. Las contraseñas se actualizan automáticamente con la ayuda del demonio updateclnt.

---

<sup>2</sup> Clog Programa para identificarse ente el servidor de identificación.

- **updateclnt:** Permite mantener actualizadas las copias de las bases de contraseñas en todos los servidores, dando la posibilidad de tener una copia original en el SCM. Para ello, el demonio *updateclnt* comprueba cada cierto tiempo si los ficheros del SCM han sido actualizados: esto hace que las actualizaciones no sean inmediatas, pues dependen del período de comprobación de *updateclnt*.

## Servicios que se ejecutan en el cliente

**venus:** Es un demonio que se ejecuta únicamente en los clientes CODA y tiene como función la de realizar dos actividades importantes:

- Dialogar con el servidor y realizar el intercambio de datos y ficheros.
- Dialogar con el controlador del kernel de la máquina local para pasarle estos datos y que genere el contenido del directorio virtual en /codas.

## Funcionamiento

- Usa RPC para habilitar las llamadas a procedimiento remoto.
- Si modifica un archivo se propaga al servidor de forma síncrona.
- Si se detecta que no hay conexión se pasa a modo desconectado.
- En este modo los cambios se almacenan en el CML (registro de modificaciones).
- Al reconectar se usa el CML para actualizar el servidor.
- Se tiene dos formas de resolución de conflictos que son:
  - Automático
  - Manual

### 1.3.2. OpenGFS

Open Global File System(GFS) es un sistema de ficheros transaccional que permite realizar o soportar el intercambio simultáneo de un dispositivo de almacenamiento de múltiples nodos de

computadoras. Esta es una manera de poner en marcha un sistema de archivos en cluster pues proporciona todos los componentes necesarios para hacerlo. (Luis Alberto Cuenca 2011)

Las máquinas y dispositivos de almacenamientos están conectados a través de una red de canal de fibra; los nodos se adjuntan a la red.

### **Estructura del sistema de archivos distribuidos**

El sistema de ficheros GFS se divide en varios grupos de recursos (RG). Estos grupos distribuyen los recursos del sistema de archivos a través de todos los almacenamientos. Existen múltiples grupos de recursos por dispositivos y pueden ser entrelistados a través de varios dispositivos. Los grupos de recursos son esencialmente mini-sistemas de archivos. GFS también tiene un superblock<sup>3</sup> que contiene información que no puede ser distribuido a través de los grupos de recursos. Esta información incluye el número de nodos montados en el sistema de archivos, mapas de bits para calcular indicadores únicos para cada nodo, el dispositivo en el que el sistema de archivos está montado y el tamaño de los bloques del sistema de archivos. (Luis Alberto Cuenca 2011)

### **i-nodos**

Un i-nodo GFS toma todo un bloque del sistema de archivos. Cada i-nodo se divide en una sección de encabezado que contiene campos i-nodo estándar y una sección de enlaces. Cada puntero apunta a un bloque indirecto que a su vez apunta a bloques de datos.

### **Bloqueos en los dispositivos**

Los bloqueos en dispositivos son mecanismos de los nodos de las computadoras para mantener la exclusión mutua de los datos del sistema de archivos distribuidos. Estos son implementados en los dispositivos de almacenamiento y se pueden acceder con un comando SCSI. El dispositivo de aplicación de los bloqueos está limitado por los comandos de bloqueo de dispositivo pues son independientes de los demás comandos SCSI. Cada bloqueo requiere cantidades mínimas de controlador de memoria de disco. Los bloqueos de dispositivos se utilizan también para ayudar a mantener la coherencia de los metadatos cuando está en caché por varios clientes. (Luis Alberto Cuenca 2011)

---

<sup>3</sup> Superblock: Se utiliza para guardar datos y almacenar los metadatos que referencien a dichos datos.

## Estado de los bloqueos

El estado de cada bloqueo se describe por un bit. Si el bit tiene el valor de 1, el bloqueo se ha adquirido y es propiedad de un nodo de la máquina. Si el bit es 0, el bloqueo está disponible para ser adquirido por cualquier nodo. La acción del comando Dlock establece el conjunto de pruebas para determinar en primer lugar si el valor del bloqueo es 1. Si el valor es 1, el comando devuelve con un estado que indica que el bloqueo ya ha sido adquirido. Si el valor es 0, Dlock establece el bloqueo a 1 y retorna un estado bueno para el indicador. (Luis Alberto Cuenca 2011)

## Consistencia y caché

La consistencia se mantiene mediante el uso de operaciones atómicas garantizadas por el dispositivo de bloqueos al modificar los datos. El protocolo SCSI describe los comandos opcionales Reserve y Release. Cuando los dispositivos de bloqueo no están implementados en un dispositivo de almacenamiento, los comandos SCSI Reserve y Release<sup>4</sup> se puede utilizar para realizar operaciones atómicas sobre los datos. Estos comandos garantizan el acceso exclusivo, pero no aportan mucho paralelismo. Con solo una reserva por dispositivo, muchas solicitudes en conflicto no tienen que esperar hasta que el dispositivo de almacenamiento este libre. En un entorno distribuido tal acceso limitado al sistema disminuye el rendimiento y los tiempos de respuesta. (Luis Alberto Cuenca 2011)

## Repositorio de almacenamiento en red.

El repositorio de almacenamiento en red (NSP) es el manejador del volumen que soporta abstracción de un único espacio de direcciones de almacenamiento unificado para los clientes de GFS. El NSP es implementado en un controlador de dispositivo en la capa superior de la base del dispositivo SCSI y en el manejador del Canal de Fibra. Este controlador traduce los espacios de direcciones lógicas del sistema de archivos a espacios de direcciones de cada dispositivo. Los subrepositorios NSPs dividen en grupos de similares tipos de dispositivos que heredarán los atributos físicos de los dispositivos y conexiones de red. (Luis Alberto Cuenca 2011)

---

<sup>4</sup> SCSI reserve or release (Liberación o reserva de SCSI): es una protección en el nivel del dispositivo que es usada por Enterprise Media Manager (EMM) para evitar compartir involuntariamente dispositivos de cinta y tener problemas posibles de pérdida de datos.

## Grupos de recursos

GFS distribuye sus metadatos a través de la red de almacenamiento en lugar de concentrar todo en un solo superblock. Múltiples grupos de recursos se utilizan para la partición de metadatos, incluyendo mapas de bits de datos y bloques de datos, en grupos separados a los clientes para así aumentar el paralelismo y la escalabilidad del sistema de archivos, evitar los cuellos de botella y reducir el tamaño medio de las típicas operaciones de búsqueda de metadatos. Uno o varios grupos de recursos pueden darse en un solo dispositivo o un único grupo de recursos pueden incluir múltiples dispositivos. Al igual que los grupos de recursos, los grupos de bloque explotan el paralelismo y escalabilidad al permitir que múltiples hilos de un solo ordenador puedan asignar y liberar bloques de datos. Los grupos de recursos de GFS permiten a múltiples clientes a hacer lo mismo. GFS también tiene un solo bloque, el superblock, que contiene un resumen de metadatos y no se distribuye a través de los grupos de recursos. (El superblock puede repetirse para mejorar el rendimiento y la facilidad de recuperación). (Luis Alberto Cuenca 2011)

Esta información incluye el número de clientes montado en el sistema de archivos, mapas de bits para el cálculo de los indicadores únicos de cada cliente, el dispositivo en el que el sistema de archivos está montado y el tamaño del bloque del sistema de archivos. El superblock contiene también un índice estático de los grupos de recursos que describe la ubicación de cada uno de estos.

## VFS Caching

Cuando la capa de VFS necesita información de la capa específica del sistema de archivos, esta hace una llamada de función hacia el sistema de archivos dependiendo de la capa para la información.

Una máquina puede cambiar los datos en el sistema de archivos sin tener que alarmar a otras máquinas. La capa VFS siempre pide a la capa del sistema de archivos específico cuando quiere información.

La capa específica del sistema de archivos siempre puede proporcionar un nivel más actualizado de los metadatos. Todos los discos de accesos pasan por capa VFS. El sistema de archivos local puede ser muy rápido, porque el VFS evita la sobrecarga de llamar a una función pues espera a que la capa específica del sistema de archivos localice y codifique la información solicitada ya que simplemente lee los datos de su propia copia.

Un caching no controlado en un sistema de archivos en red, especialmente en un sistema de archivos de disco compartido, puede dar lugar a incoherencias de datos entre las máquinas.

## **Mejoras en el sistema de ficheros**

Se han introducido mejoras en el sistema de archivos y metadatos. Estos cambios, proporcionan aumento en la escalabilidad de GFS.

## **Consistencia de GFS**

Hay que tener cuidado cuando los metadatos son accedidos y actualizados, ya que si el Dlock no actúa en el momento oportuno, puede resultar muy fácil la corrupción de los datos y metadatos.

Este nuevo bloqueo también ha aumentado las posibilidades de estancamiento. Hay muchos lugares donde el sistema de archivos debe mantener de dos o más Dlock para realizar una operación. Por ejemplo, la operación de búsqueda requiere simultáneamente dos bloqueos. La operación de búsqueda toma un directorio y el nombre de un archivo en ese directorio y devuelve el i-nodo para ese archivo. Dos bloqueos se requieren para esta operación: un bloqueo se realiza cuando el directorio es leído y cuando el número i-nodo del archivo es determinado. El otro bloqueo se realiza al tiempo que el i-nodo se lee. Estos dos bloqueos deben ser realizados, al mismo tiempo. (Luis Alberto Cuenca 2011)

## **Buffer Caché**

El buffer de memoria caché es un componente importante de los modernos sistemas operativos UNIX, porque evita el exceso de accesos a disco, ya que el sistema operativo guarda los bloques de disco utilizados recientemente en una sección de la memoria llamada “buffer caché”. Las futuras solicitudes de datos que ya están en el buffer de memoria caché se pueden completar con rapidez, ya que no se requiere el acceso a disco.

Si los datos solicitados no están en el buffer de memoria caché, estos se leen desde el disco, y a continuación copia en un buffer de memoria caché, así como para el programa de usuario. Esto se aplica tanto a los metadatos de archivos y bloques. En este caso el rendimiento es mucho mayor al utilizar el buffer de caché en vez de acceder al disco. La caché de bloques de metadatos también mejora el rendimiento para archivos de gran tamaño a causa de las repetidas solicitudes indirectas de referencias de bloques. El uso de caché en el buffer de GFS es complicada por la capacidad de acceder a múltiples clientes y caché de los bloques de disco.

## **Administración del espacio en disco**

La administración del espacio libre en GFS se basa en el enfoque de mapa de bits. Por cada bits del sistema de fichero en un determinado grupo de recursos, hay un único bit para representar si el bloque está libre o no. La utilización de un esquema de base extendido, puede costar más en términos de espacio, pero proporciona un mejor rendimiento. En lugar de hacer el seguimiento de cada bloque del sistema de archivos en un grupo de recursos, se limita a los bloques libres. Por cada grupo de bloques libres del sistema de archivos en un grupo de recursos habrá una medida que no pierde de vista el punto de partida y el número de bloques en el grupo. (Luis Alberto Cuenca 2011)

## **Recuperación de error**

La recuperación de error es importante en un sistema de archivos de discos compartidos, porque ayuda a identificar donde está fallando y así poder controlar esta falla a tiempo.

Todos los clientes manipulan directamente los metadatos, por lo que el fracaso de cualquier cliente puede dejar los metadatos en un estado inconsistente. Además, como hay tantas máquinas accediendo a los discos, no es práctico para todos ellos desmontar y esperar a comprobar un sistema de ficheros (fsck) para completar cada vez que un cliente falla. Es importante que las incoherencias causadas por un cliente sean localizadas y fácilmente reparadas mientras el sistema de ficheros está en funcionamiento. (Luis Alberto Cuenca 2011)

### **1.3.3. GlusterFS**

GlusterFS es un sistema de archivos en cluster capaz de escalar a peta-bytes y manejar miles de clientes. GlusterFS se puede combinar de forma flexible con productos físicos, virtuales y recursos de la nube para ofrecer almacenamiento empresarial con alta disponibilidad y rendimiento a una fracción del costo de las soluciones tradicionales.

GlusterFS está diseñado para trabajo de alto rendimiento y entornos de nube virtualizada. A diferencia de los centros de datos tradicionales, los entornos cloud requieren multialquiler, junto con la capacidad de aumentar o disminuir los recursos bajo demanda.

La ampliación de la capacidad, rendimiento y disponibilidad, se la puede hacer con el uso de nubes públicas y entornos híbridos, ya no hay dependencia de un proveedor específico.

## Principales consideraciones del diseño de GlusterFS

Para el diseño de GlusterFS se tomaron en cuenta algunos factores como los que se citan a continuación:

- **Capacidad de expansión:** La expansión de GlusterFS es muy grande, ya que fácilmente puede sobrepasar un peta-byte<sup>5</sup> de información.
- **Facilidad de administración:** Utiliza un esquema parecido al NFS (no NFS como tal) debido a que GlusterFS implementa ciertas características de seguridad a la información que va a replicar. La entrada y salida de información no está vinculada a perfiles de hardware y sistema operativo.
- **Fiabilidad:** Posee un almacenamiento incremental ya que no deja de guardar información.
- **Escalabilidad de flujo I/O<sup>6</sup>:** Manejador para agrupaciones de I/O.

## Ventajas sobre el transporte RDMA<sup>7</sup> y características

- Escalabilidad sobrepasando peta-bytes.
- Soporte para Infiniband, RDMA y TCP/IP.
- Balanceo de carga.
- Alta disponibilidad.
- Completamente distribuido.
- Replicación de los archivos.
- Encriptación de los datos.

---

<sup>5</sup> Peta-bytes: Un **peta-byte** es una unidad de almacenamiento de información cuyo símbolo es el **PB**, y equivale a  $10^{15}$  bytes.

<sup>6</sup> I/O (Entrada y salida); hace referencia a la entrada y salida de la información en un medio.

<sup>7</sup> RDMA Remote Direct Memory Access permite mover datos directamente desde la memoria de un computador a otro. Esto permite alto rendimiento y baja latencia de red.

- Diseñado para trabajar en ambientes cluster.
- Comunicación segura con los nodos, ya que implementa mecanismos de autenticación.
- La implementación es fácil.
- Posee soporte.
- Posee licencia GPL.

## Nuevas características de GlusterFS3. 2

- **Geo-replication:** ofrece un servicio de replicación continua, asíncrona e incremental de un sitio a otro sobre redes de área local (LAN), redes de área amplia (WANs) y a través de Internet. Con la utilización de Gluster Geo-replication, se puede establecer redundancia de datos, proporcionando una recuperación ante desastres a través de redes LAN, WAN y conexiones a Internet existentes.
- **Directory Quota:** permite establecer cuotas de uso de espacio en disco, directorios o volúmenes. Los administradores pueden controlar el uso de espacio en disco y/o nivel de volumen en GlusterFS mediante la asignación de espacio en disco a cualquier nivel en el volumen y jerarquía de directorios.
- **Top and Profile:** permite controlar los diferentes parámetros de la carga de trabajo, lo que ayuda en la planificación de la capacidad y las tareas de optimización del rendimiento del volumen.
- **POSIX ACLs Support:** permite asignar diferentes permisos para distintos usuarios o grupos a pesar de que no se corresponden con el propietario original o el grupo propietario.

## Componentes GlusterFS3. 2

Para la utilización de GlusterFS se hacen uso de dos componentes principales que son:

- **GlusterFS3. 2 Servidor**

Aquí se exportan los volúmenes para los nodos del cluster, se pueden configurar varios GlusterFS3. 2 Server para evitar tener un punto único de fallos. El demonio que se utiliza en el servidor es: **glusterfsd**. El archivo de configuración se encuentra en la siguiente dirección: **/etc/glusterfs/glusterfs-server.vol**

- **GlusterFS3.2 Cliente**

Este componente es instalado y configurado en los nodos del cluster, la función que cumple es la de montar los volúmenes que se encuentran configurados en el servidor. El demonio que se utiliza es el siguiente: **glusterfs**. El archivo de configuración se encuentra en la siguiente dirección: **/usr/local/etc/glusterfs/glusterfs-client.vol**. Un aspecto importante a tener en cuenta en GlusterFS es que los clientes deben tener soporte para fuse<sup>8</sup>.

### Consola de administración del Gluster

Es una utilidad de línea de comandos que simplifica la configuración y gestión del entorno de almacenamiento. Con el uso de esta consola los administradores pueden crear nuevos volúmenes, iniciar y detener los volúmenes, según como se requiera. Los administradores también pueden utilizar los comandos para crear scripts que posibiliten la automatización, así como utilizar los comandos como un API para permitir la integración con aplicaciones de terceros.

### Propiedades de GlusterFS

GlusterFS posee propiedades que permiten mejorar el rendimiento del sistema de archivos. Estas propiedades están clasificadas de la siguiente manera:

*Tabla 1 Propiedades de GlusterFS.*

Propiedad	Elementos
Rendimiento	read ahead write behind threaded I/O IO-cache stat pre-fetch booster
Replicación	afr
Clustering	stripe unify
Balanceo de carga	ALU NUFA Random Round Robin

---

<sup>8</sup> Fuse: Módulo para el kernel, ese es un soporte para GlusterFS.

	Switch
Depuración	trace
Características extras	Filter posix-locks trash fixed-id
Protocolos servidor	tcp/server ib-sdp/server ib-verbs/server
Protocolos cliente	tcp/server ib-sdp/server ib-verbs/server
Autenticación	auth.ip auth.login
Encriptación	rot-13

### Rendimiento

Las propiedades de rendimiento trabajan bien cuando son cargadas tanto del lado del servidor como del cliente.

#### **read-ahead**

Esta propiedad realiza una pre-búsqueda avanzada de una secuencia de bloques de datos basada en predicción. Es usada en el momento en que la aplicación está ocupada procesando datos que ha leído con anterioridad, para que GlusterFS pueda realizar una pre-lectura del siguiente batch de datos y dejarlo listo para utilizarlo, es decir al momento de leerse un bloque de información, en ese mismo instante también está realizando una pre-lectura del próximo bloque, de este modo las lecturas son más rápidas y adicionalmente cambia el comportamiento como si fuera un canalizador de lectura.

Dentro de read-ahead hay una opción page-size que indica el tamaño de un bloque y page-count describe la cantidad de bloques a ser pre-buscados.

#### **write-behind**

Generalmente la opción de escritura es más lenta que la de lectura, es por ello que se utiliza esta propiedad para mejorar significativamente el rendimiento de escritura usando una técnica de escritura en segundo plano a la que se conoce como background. De este modo múltiples operaciones de escritura son agregados sin limitación alguna y son escritas en un segundo plano. La opción `aggregate-size` de `write-behind` determina el tamaño del bloque a ser escrito. Se recomienda realizar un benchmark con un incremento en el valor del rango de `aggregate-size` para así obtener un valor óptimo y lograr un mejor rendimiento.

## **io-threads**

Agrega sincronización para la funcionalidad de lectura/escritura. Esta propiedad hace que se usen de una mejor manera todos los recursos.

## **io-cache**

Ayuda a reducir la carga en los servidores.

## **Replicación**

GlusterFS trae consigo la propiedad `afr` que permite la replicación de información.

**afr** (Automatic – File – Replication) replicación automática de archivos; con esta propiedad se evita tener un punto único de fallo, ya que con esto se pueden crear varios espejos.

## **Cluster**

Como GlusterFS está diseñado para trabajar en un ambiente de cluster, cuenta con las siguientes características: `stripe` y `unify`.

## **stripe**

Clasifica los archivos de entrada de acuerdo con un patrón definido por el usuario dentro de un tamaño de bloque dado.

## **unify**

Combina varios nodos de almacenamiento en un solo gran nodo de almacenamiento en el servidor.

## **Balanceo de carga**

Permite decidir cómo se va a distribuir la creación de nuevas operaciones sobre el sistema de archivos del cluster tomando en cuenta la carga y la disponibilidad.

A continuación se describirán cada una de las propiedades que ayudan al balanceo de carga.

## **ALU**

Es uno de los manejadores más avanzados disponibles en GlusterFS, este balancea la carga a través de volúmenes, tomando varios factores en cuenta.

## **NUFA**

Permite manejar distintos sistemas de ficheros sin importar que sean de un mismo cluster.

## **Round-Robin**

Crea archivos al estilo round-robin. Cada cliente tiene su propio bucle round-robin. Es una buena opción el utilizarlo cuando los archivos son muy similares tanto en tamaño como en patrones de acceso de I/O. RR chequea el espacio libre en el servidor antes de ejecutarse, lo que ayuda a conocer cuándo agregar otro bloque en un servidor.

## **Depuración**

GlusterFS ofrece la opción de realizar una depuración con el fin de poder encontrar posibles errores. Para ello cuenta con la propiedad trace.

## **trace**

Hace posible la producción de gran cantidad de información para fines de depuración: esta información es escrita en un archivo de log que es propio de GlusterFS y que se encuentra en **`/var/log/gluster/glusterfs.log`**

## **Características extras**

Estas características extras permiten aumentar la funcionalidad de GlusterFS, ya que permite ampliar las utilidades que ofrece. Una de estas características muy importante es posix-locks con la cual se logra establecer un bloqueo, propiedad que debería ser utilizada por los archivos que van a ser accedidos y modificados muchas veces.

Además posee otras características que se detallan a continuación:

### **filter**

Permite un filtrado avanzado basado en el nombre de archivo y/o atributo. Hasta la versión actual solo se permite exportar en modo read-only.

### **Fixed-id**

Muestra el id del usuario y grupo, especificado para todos los archivos y carpetas.

## **Almacenamiento**

### **posix**

GlusterFS confía en sistemas de archivos basados en discos tales como ext3 o XFS para manejar la administración de bloqueo del dispositivo, entonces la función que cumple posix es la de unir el GlusterFS server con el sistema de archivos basado en disco.

## **Autenticación**

GlusterFS permite la autenticación entre los nodos por medio de IP así como también utilizando usuario y password.

## **Encriptación**

Esta característica es muy importante ya que con esta se logra tener seguridad de la información que se está transmitiendo entre los diferentes nodos del cluster.

### **Rot-13**

Provee una encriptación y desencriptación de archivos utilizando el algoritmo rot-13, con lo que se logra proteger la información que se está transmitiendo entre cada uno de los nodos del cluster.

### 1.4. Sistema de archivos distribuidos para la solución.

Varias son las implementaciones (FOSS) de sistemas de archivos distribuidos. En las tabla 2 y 3 se describe una relación comparativa que incluye los exponentes más relevantes de los sistemas de archivos y sus características fundamentales.

*Tabla 2 Características relevantes de los sistemas de archivos distribuidos*

No	Característica	Valor predeterminado
1	Replicación de archivos a varios equipos	9
2	Encriptación de la información	9
3	No se posee un punto único de montaje	9
4	Soporte bloqueo de archivos	9
5	Punto único de fallo	8
6	Soporte (Kernel)	8
7	Balanceo de carga	8
8	Escalabilidad para transferir grandes cantidades de datos	8
9	Diseño para ambientes de cluster	7
10	Trabajo en modo desconectado	7
11	Caching local	7
12	No requiere de un servidor exclusivo para la implementación	7
13	Licencia GPL	7
14	Control de cuotas del diseño	7

Tabla 3 Comparación de los sistemas de archivos distribuidos.

Características	Sistemas de replicación de archivos		
	CODA	OpenGFS	GlusterFS
1	9	9	9
2	9	0	9
3	0	9	9
4	0	9	9
5	8	8	8
6	8	0	8
7	0	0	8
8	0	0	8
9	0	0	7
10	7	0	0
11	7	7	0
12	0	7	7
13	7	7	7
14	0	0	7
<b>Total</b>	55	56	96

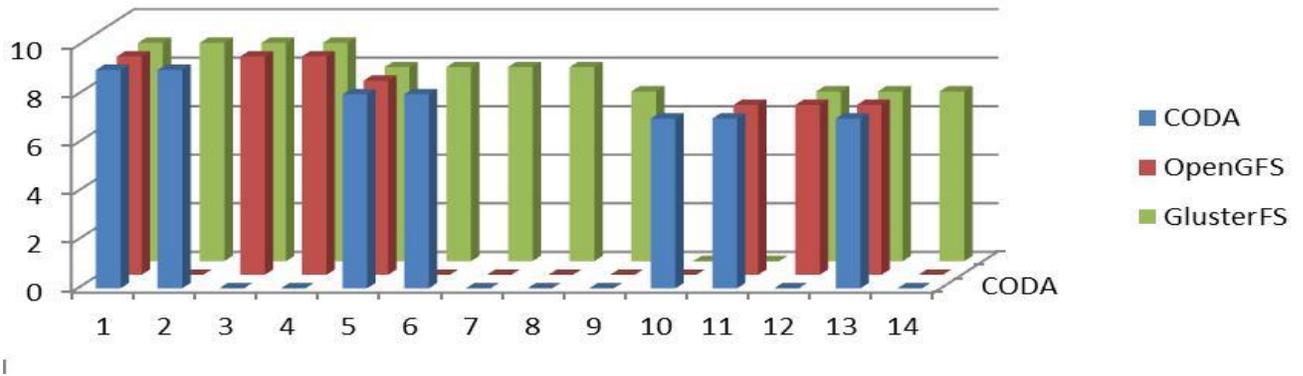


Figura 1 Representación de los resultados arrojados.

La tabla 2 muestra un resumen de las características más importantes de los sistemas de archivos distribuidos; se le asignaron valores a las características en dependencia del nivel de importancia que se le atribuye a los sistemas de archivos distribuidos. Luego se elabora la tabla 3, para la asignación de los puntos en dependencia del sistema de archivos distribuido obteniendo un total de puntos mediante la presencia de la característica y el valor que le corresponde; dando la posibilidad de escoger el mayor puntaje, demostrando así la mejor opción de selección; quedando constatado como mejor sistema de archivo distribuido GlusterFS3.2 con un total de 96 puntos.

## 1.5. Marco de desarrollo

Zentyal es una plataforma de red unificada para las PYMEs, conocida anteriormente con el nombre de eBox Platform. Puede actuar gestionando la infraestructura de red, las amenazas de seguridad, como puerta de enlace a Internet (Gateway), como servidor de oficina, como servidor de comunicaciones unificadas o una combinación de estas. Incluye, además un marco de desarrollo para facilitar el desarrollo de nuevos servicios basados en Unix. Su propietario y patrocinador es la empresa española eBox Technologies S.L.

Mediante la interfaz de programación de aplicaciones (API) que brinda Zentyal se pueden automatizar fácilmente sus tareas. Para lograrlo solamente se necesitan algunos conocimientos básicos del lenguaje de programación Perl, además de tener conocimiento de los métodos que se quiere utilizar. Una ventaja que se tiene con esto es el hecho de que la interfaz web emplea la misma interfaz de programación.

Zentyal está diseñado precisamente pensando en la extensibilidad y es relativamente sencillo crear nuevos módulos. Cualquiera con conocimientos del lenguaje Perl puede aprovecharse de las facilidades que proporciona Zentyal para la creación de interfaces Web, y también beneficiarse de la integración con el resto de módulos y demás características comunes de Zentyal.

El diseño de Zentyal es completamente orientado a objetos y hace uso del patrón Modelo-Vista-Controlador (MVC), de forma que el desarrollador sólo necesita definir qué características desea en su modelo de datos, y el resto será generado automáticamente por Zentyal. Además se dispone de una herramienta llamada `zmoddev` que facilita el desarrollo de nuevos módulos, proporcionando plantillas auto-generadas en función de parámetros definidos por el usuario, lo que se traduce en un ahorro de tiempo.

Este proceso se puede llevar a cabo a través de una máquina virtual o usando la propia máquina; el primer caso es recomendado en virtud de no tener que pasar por situaciones no deseadas. Para el mismo se debe contar con los siguientes aspectos:

- Se puede utilizar VirtualBox3 (aunque esa es una decisión muy personal) con una instalación limpia de Zentyal 2.2.
- Probablemente se quiera establecer una conexión SSH a la máquina virtual para trabajar más cómodamente.
- Hay que tener todos los paquetes necesarios de construcción que ya están en los repositorios o crear un repositorio en la propia máquina con los paquetes y dependencias necesarias.
- Siempre es una buena práctica utilizar las capacidades de instantáneas, permitiendo mantener un ambiente limpio para volver cuando sea necesario.

Luego de hacer la elección se procede a la instalación de la herramienta de desarrollo del módulo de Zentyal (también se instalará el paquete **zbuildtools** necesario para construir módulos de Zentyal):

```
sudo apt-get install zmoddev
```

Para crear el módulo hay que utilizar `zmoddev`, una colección de scripts de conveniencia que ayudará a crear la estructura y los archivos básicos necesarios en un módulo de Zentyal. Para la generación de las plantillas auto-generadas hay que ejecutar el siguiente comando:

```
zentyal-moddev-create --module-name nombre -main-class NombreClase --version 1.0
```

Lo anterior creará un directorio llamado `nombre` con todos los archivos que componen un módulo de Zentyal. El modelo de datos para gestionar los módulos es muy sencillo: se necesitará un campo para almacenar el nombre del módulo y otro campo para habilitar y deshabilitarlo.

Se utilizará `zentyal-moddev-model`, una herramienta que permite añadir fácilmente nuevos modelos a nuestro módulo de Zentyal. Luego de tener bien claros los modelos con que contará el módulo se procede a su creación mediante la ejecución del siguiente comando dentro del directorio `nombre`:

```
zentyal-moddev-model --main-class NombreClase --name NombreModelo --field module: Select -field enabled: Boolean -field path: Text --model table
```

El comando anterior suma un nuevo modelo con el nombre deseado (en este caso el nombre es `NombreModelo`) que se compone de un tipo `Select` llamado `module`, un tipo `Boolean` llamado `enabled` y uno `text` llamado `path`. El tipo de modelo es una tabla en lugar de un formulario como se hizo en la versión 1.0. `main-class` es `NombreClase` como en la creación inicial del módulo.

Es importante que se sepa que existen diferentes tipos de modelos en Zentyal como `Txt`, `Int`, `Boolean`, `Select`, `Port`, `Password`, `PortRange`, `Service`, `HostIP`, `Host`, `IPAddr`, entre otros. Por ahora no hay ninguna entrada en el menú para ver la vista asociada a este módulo, pero se puede fácilmente conocer la dirección URL. Por ejemplo la dirección del modelo `NombreModelo` que se creó se encuentra ubicado en la siguiente dirección: `nombre/src/EBox/NombreClases/Model` aquí es donde tiene mayor fuerza el código. Luego de tenerlo todo listo se corren los siguientes comandos vía consola:

```
cd nombre (para situarse en el directorio) y zentyal-package para construir el paquete Debian, que se encontrará en el directorio nombre/debs-ppa. Entonces se procede a la instalación, la cual puede hacerse dando doble clic sobre él .deb o usando dpkg mediante la línea de comandos sudo dpkg -i debs-ppa/zentyal-nombre_1.0.deb.
```

De esta forma Zentyal da la posibilidad de integrarle nuevos módulos. Para ello se hará uso de la metodología de desarrollo de software OpenUP, pues esta es la utilizada por el proyecto Nova para Servidores siendo la herramienta CASE (Computer Aided Software Engineering, en español Ingeniería de Software Asistida por ordenador) a utilizar Visual Paradigm que utiliza UML (Unified Modeling Language, en español Lenguaje Unificado de Modelado)” como lenguaje de modelado. Además posee características gráficas muy cómodas que facilitan la realización de los diagramas de modelado. Permite especificar, analizar y diseñar el sistema.

Para la implementación de la solución se hará uso del lenguaje de programación Perl v5.10, puesto que la herramienta a la que se le integrara el módulo está desarrollada en Perl. Además se hará uso del editor de textos Gedit, que viene instalado por defecto en el entorno de escritorio GNOME, es una aplicación potente en la edición de documentos de texto así como la edición de código para programadores. Uno de sus puntos fuertes es que pueden añadirsele plugins y complementos para personalizar las capacidades del software.

## **1.6. Conclusiones del capítulo**

A lo largo de este capítulo se ha explicado y detallado la base sobre la que se trabajará en aras de desarrollar la aplicación. Del estudio se definió lo siguiente: trabajar sobre el sistema de archivos distribuido GlusterFSv3.2 permitiendo aprovechar características que no estaban presente en los demás sistemas de archivos distribuidos. Para el desarrollo del módulo se hace uso del lenguaje de programación Perl, permitiendo utilizar las potencialidades que la plataforma brinda, logrando una fácil integración de los módulos, pues la misma esta implementada en el lenguaje Perl v5.10. Se utiliza la metodología de desarrollo OpenUP dando la posibilidad de guiar el desarrollo del módulo a través de un proceso iterativo, completo y extensible.

### Capítulo 2.- Análisis y diseño del sistema.

#### 2.1. Introducción

En este capítulo se define el análisis y diseño de la solución utilizando la metodología de desarrollo OpenUp.

OpenUp es un proceso iterativo del desarrollo del software que es mínimo, completo, y extensible. Es un proceso para equipos de desarrollo pequeños y que le dan valor a la colaboración y a las necesidades de los stakeholder. (Daniel Levano 2008)

Esta metodología expone que lo primero que se debe hacer es la definición de los actores del sistema, posteriormente los requisitos funcionales para la modelación del Diagrama de Casos de usos del Sistema, y los requisitos no funcionales. Finalmente propone la modelación del Diagrama de Clases del Diseño y el Diagrama de Interacción por cada caso de uso.

#### 2.2. Actores del sistema

Los actores son las diferentes personas que interactúan con el sistema dentro del contexto de la función y el comportamiento que se describirá. Un actor es algún elemento que se comunica con el sistema y que es externo al sistema.

Por la frase "interactúa con el sistema" se debe entender que el actor envía o recibe del sistema unos mensajes. En pocas palabras, el actor lleva a cabo los casos de uso.

*Tabla 4 Actores del sistema*

Actor del sistema	Descripción
Administrador	El administrador es la persona que va a interactuar con el sistema para establecer la implantación del sistema de archivo distribuido.

### 2.3. Requisitos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, no alteran la funcionalidad del producto, por lo que se mantienen invariables sin importar con que propiedades o cualidades se relaciona. (Maikel Ramírez, Millán 2011)

A continuación se listan los requisitos funcionales por cada caso de uso. Los casos de uso van a estar representados por las siglas CU precedido del número que ocupa en la lista.

#### ➤ **CU1: Gestionar Servidores**

- Adicionar servidor
- Mostrar servidores
- Eliminar servidor
- Modificar servidor
- Verificar que esté conectado

#### ➤ **CU2: Insertar Volumen**

- Seleccionar el tipo de transporte
- Seleccionar el tipo de volumen
- Insertar la cantidad base de bloques

#### ➤ **CU3: Insertar Bloque**

- Seleccionar el servidor de la lista de servidores, estando esté conectado
- Insertar la dirección del bloque

#### ➤ **CU4: Modificar Volumen**

- Poder modificar los datos previos del volumen
- **CU5: Modificar Bloque**
  - Modificar los datos previos del bloque
- **CU6: Mostrar volumen**
  - Mostrar los volúmenes existentes
- **CU7: Mostrar Bloque**
  - Mostrar los bloques existentes
- **CU8: Eliminar Volumen**
  - Eliminar el volumen de la lista de volúmenes
- **CU9: Eliminar Bloque**
  - Eliminar el bloque de la lista de bloques
- **CU10: Administrar servicio de Volumen**
  - Detener el volumen
  - Iniciar el volumen
  - Reiniciar el volumen
- **CU11: Administrar Servicio General**
  - Detener el servicio de GlusterFSv3.2
  - Iniciar el servicio de GlusterFSv3.2
  - Reiniciar el servicio de GlusterFSv3.2
- **CU12: Instalar Firewall**

- Instalar firewall
- **CU13: Administrar Plugin**
  - Instalar plugin
  - Desinstalar plugin
- **CU14: Instalar Antivirus**
  - Instalar antivirus

### 2.4. Requisitos no funcionales

Los requisitos no funcionales son los requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, si no a las propiedades emergentes de este. (Maikel Ramírez, Millán 2011)

- Apariencia o interfaz externa
  - El producto debe ser legible y con los colores establecidos en el manual de identidad del proyecto.
  - Los mensajes mostrados al usuario deben seguir los patrones definidos por Zentyal.
- Usabilidad
  - El administrador podrá permanecer en el sistema el tiempo que estime conveniente.
  - Se utiliza el idioma español para los mensajes y textos de la interfaz.
  - El funcionamiento del sistema será intuitivo y requerirá de conocimientos mínimos para su uso.
- Accesibilidad

- La información y las funcionalidades estarán disponibles y el administrador podrá acceder a ellas en todo momento.

### ➤ Disponibilidad

- Una vez que el sistema esté publicado estará siempre disponible y con la información que solicite el administrador, dependiendo únicamente de la operatividad del Zentyal.

### ➤ Rendimiento:

- La aplicación permitirá que múltiples usuarios estén conectados a la vez.
- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 10 segundos para las recuperaciones.

### ➤ Legales:

- Las herramientas seleccionadas para el desarrollo del producto están respaldadas por licencias libres, bajo las condiciones de software libre. Para la herramienta Visual Paradigm, la cual no es libre se utiliza la licencia que posee la Universidad.
- La aplicación y toda la documentación generada pertenecen al grupo de proyecto Gestión Documental y Archivística de la Universidad de las Ciencias Informáticas.

### ➤ Software:

- Las PC servidor deben tener instalado el navegador web Mozilla Firefox.
- La PC servidor debe contar con el servidor web Apache 2.0.
- La PC servidor debe tener instalado el sistema de archivos distribuidos GlusterFSv3.2

➤ Hardware:

### **Procesamiento**

- El servidor de aplicaciones requiere de 1 CPU Intel Pentium 4 o compatible para su correcto funcionamiento.

### **Memoria RAM:**

- El servidor de aplicaciones requiere de 2 Gb de memoria RAM para su correcto funcionamiento.

### **Capacidad en disco**

- El servidor de aplicaciones requiere como mínimo de 500 Gb disponibles para su correcto funcionamiento.

### **Soporte**

- Realizar pruebas y mantenimiento necesarios para lograr el mejoramiento y evolución en el tiempo.

➤ Restricciones de diseño e implementación:

- El sistema se desarrollará utilizando como lenguaje de programación Perl.
- El servidor debe tener instalado servidor web Apache 2.0.
- Las vistas deben ser desarrolladas en html.
- Para la modelación del sistema se utilizará Visual Paradigm 8.0.
- La metodología de desarrollo de software es OpenUp, usando el lenguaje de modelación UML.

## **2.5. Diagrama de caso de uso del sistema**

El diagrama de casos de uso del Sistema es un diagrama de comportamiento, que muestra la relación entre los actores y los casos de uso del sistema; en sí representa la funcionalidad que ofrece el sistema en lo referido a su interacción externa. En el diagrama de casos de uso se representa el sistema como

una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea de interacción.

Los casos de uso “son fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia. (Ivar Jacobson 2000)

La figura 2 muestra el diagrama de casos de uso, el mismo va a estar compuesto por el administrador inicializando los casos de uso definidos anteriormente excepto para los casos de uso Insertar Bloque que va a contener una relación de inclusión con el caso de uso Insertar Volumen y una relación extendida con el caso de uso Modificar Volumen. Los casos de uso Modificar Bloque y Mostrar Bloque van a contener una relación extendida de los casos de uso Modificar Volumen y Mostrar Volumen.

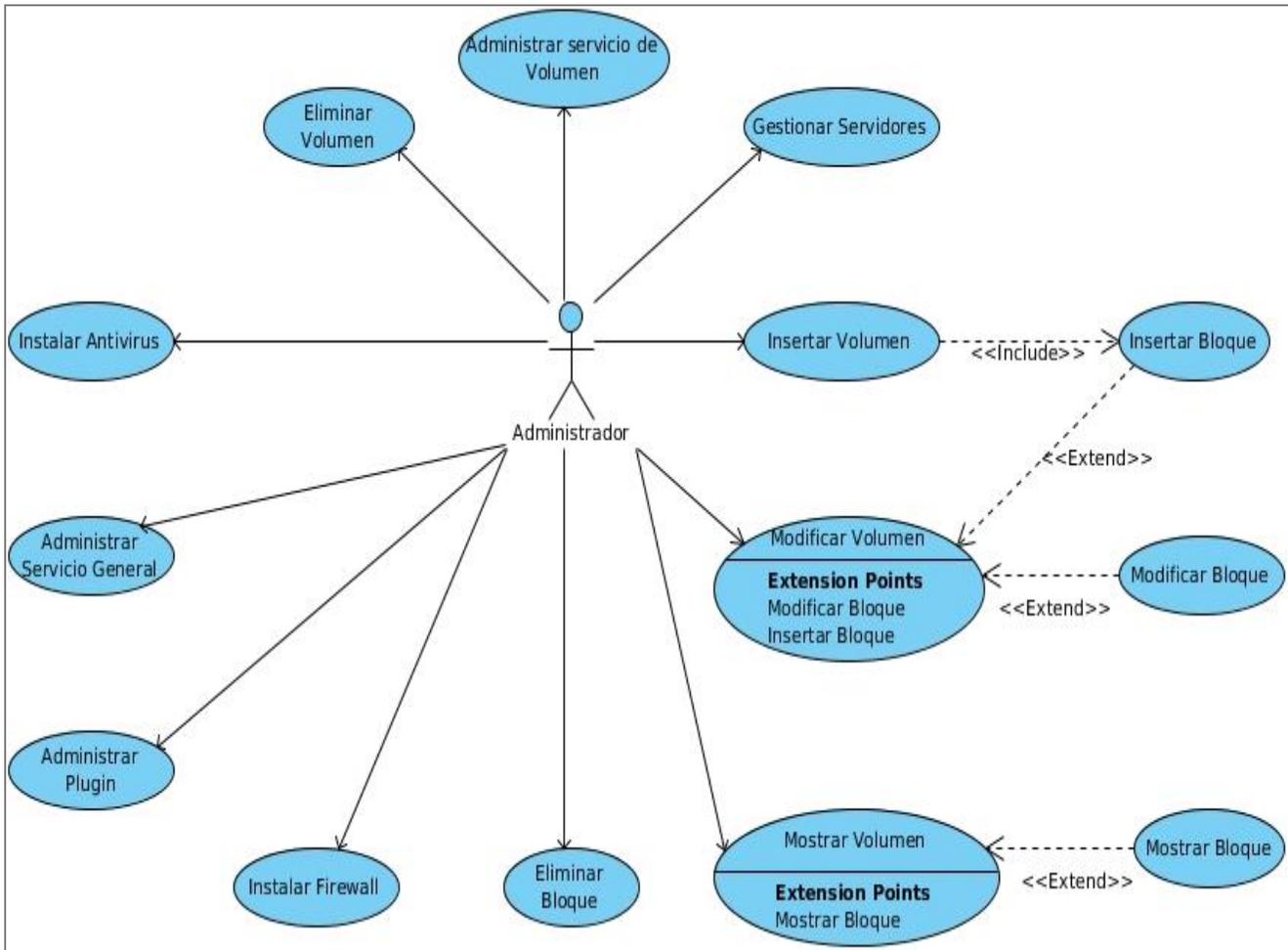


Figura 2 Diagrama de casos de uso del sistema

## 2.6. Descripción de los Casos de uso del sistema

Los casos de usos del sistema y los requisitos funcionales constituyen la entrada fundamental para el diseño del análisis del sistema. A continuación se describen los casos de uso críticos del sistema.

### 2.6.1. Caso de uso Gestionar Servidores

Tabla 5 Descripción del Caso de uso Gestionar Servidores

Caso de uso	Gestionar Servidores
Objetivo	Permite al administrador gestionar (adicionar, eliminar y modificar) los servidores disponibles.

<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción “Servidores”, luego el sistema muestra la interfaz Gestionar Servidores, el usuario selecciona la opción Gestionar dando la posibilidad de adicionar, eliminar o modificar según lo que desee hacer, luego introduce los datos y el sistema da la opción de guardar los cambios hechos.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal.	
<b>Postcondiciones</b>	Nuevo servidor registrado en el sistema, modificado o eliminado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Acción del sistema</b>	
1. Selecciona la opción Servidores.	2. Muestra la interfaz Gestionar Servidores.	
3. Selecciona la opción Gestionar.	4. Muestra un listado con todos los servidores disponibles.	
5. Selecciona una opción. a) Si selecciona la opción añadir servidor véase la sección Adicionar servidor. b) Si selecciona la opción modificar servidor véase la sección Modificar servidor. c) Si selecciona la opción eliminar servidor véase la sección Eliminar servidor		
<b>Sección “Adicionar servidor”</b>		
<b>Acción del actor</b>	<b>Acción del sistema</b>	
	1. Muestra los campos para introducir los datos del nuevo servidor.	
2. Introduce los datos del nuevo servidor.	3. Verifica que los datos introducidos sean correctos. 4. Si hay errores en los datos véase el flujo alternativo 4.1	

	5. Si los datos son correctos el sistema inserta el nuevo servidor terminando así el caso de uso.
<b>Flujo Alterno 4.1</b>	
	4. 1 Si hay errores en los datos del nuevo servidor, el sistema muestra un mensaje de error y retorna al paso 1.
<b>Sección “Modificar servidor”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
	1. Muestra los datos actuales del servidor seleccionado que pueden ser modificados.
2. Modifica los datos del servidor seleccionado.	3. Verifica que los datos introducidos son correctos.
	4. Si hay errores en los datos véase el flujo alternativo 4.1
	5. Si los datos fueron introducidos correctamente, el sistema modifica los datos seleccionados terminando así el Caso de uso.
<b>Flujo Alterno</b>	
	4.1 Muestra un mensaje de error y retorna al paso 3.
<b>Sección “Eliminar servidor”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
	1. Elimina el servidor seleccionado terminando así el Caso de uso.

### 2.6.2. Caso de uso Insertar Volumen

*Tabla 6 Descripción del Caso de uso Insertar Volumen*

Caso de uso	Insertar Volumen
<b>Objetivo</b>	Permite al administrador adicionar un volumen.
<b>Actores</b>	Administrador(Inicia)
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción de Volúmenes después da en la opción Gestionar y el administrador dando clic en Adicionar nuevo, introduce los datos y el sistema da la opción de guardar los cambios efectuados.
<b>Complejidad</b>	Alta

<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Cuando se inserte un volumen para que este exista tiene que obligatoriamente tener un bloque.	
<b>Postcondiciones</b>	Nuevo volumen registrado en el sistema.	
<b>Flujo Normal de Eventos</b>		
	<b>Acción de actor</b>	<b>Acción del sistema</b>
	1. Selecciona la opción Volúmenes.	2. Muestra la ayuda de volúmenes y da la posibilidad de Gestionar.
	3. Selecciona la opción Gestionar.	5. El sistema muestra los campos para introducir los datos.
	4. Selecciona la opción adicionar nuevo.	7. El sistema verifica que los datos introducidos sean correctos.
	6. Introduce los datos.	8. Si existen errores véase el flujo alternativo 8.1
		9. Si los datos fueron introducidos correctamente, el sistema inserta los datos seleccionados terminando así el Caso de uso.
<b>Flujos Alternos</b>		
		8.1 Muestra un mensaje de error y retorna al paso 5.

### 2.6.3. Caso de uso Insertar Bloque

Tabla 7 Descripción del Caso de uso Insertar Bloque

<b>Caso de uso</b>	<b>Insertar Bloque</b>
<b>Objetivo</b>	Permite al administrador insertar un bloque
<b>Actores</b>	Administrador(Inicia)
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Bloques de un volumen, luego da la opción de adicionar un nuevo bloque permitiendo introducir los datos. Después el sistema da la opción de guardar los cambios efectuados.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen.

<b>Postcondiciones</b>	Nuevo Bloque registrado en el sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Acción del sistema</b>	
1. Selecciona la opción Bloques de un volumen.	2. El sistema muestra un listado con todos los bloques disponibles para el volumen.	
3. Selecciona la opción adicionar nuevo.	4. Muestra los campos para introducir los datos.	
5. Introduce los datos.	6. El sistema verifica que los datos introducidos sean correctos.	
	7. Si existen errores véase el flujo alterno 7.1	
	8. Si los datos fueron introducidos correctamente, el sistema inserta los datos seleccionados terminando así el caso de uso	
<b>Flujos Alternos</b>		
	7.1 Muestra un mensaje de error y retorna al paso 4.	

### 2.6.4. Caso de uso Modificar Volumen

*Tabla 8 Descripción del Caso de uso Modificar Volumen*

<b>Caso de uso</b>	<b>Modificar Volumen</b>	
<b>Objetivo</b>	Permite al administrador modificar un volumen	
<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción de Volúmenes después da en la opción Gestionar y el administrador dando clic en modificar, introduce los datos y el sistema da la opción de guardar los cambios efectuados.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen.	
<b>Postcondiciones</b>	Volumen modificado en el sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Acción del sistema</b>	
1. Selecciona la opción Volúmenes.	2. Muestra la ayuda de los volúmenes y la opción Gestionar.	

<ol style="list-style-type: none"> <li>3. Selecciona la opción Gestionar.</li> <li>5. Selecciona la opción modificar volumen del listado de volúmenes.</li> <li>7. Modifica los datos.</li> <li>8. Introduce los datos modificados,</li> </ol>	<ol style="list-style-type: none"> <li>4. Muestra el listado de volúmenes.</li> <li>6. El sistema muestra los campos con los datos para que estos puedan ser modificados.</li> <li>9. El sistema valida los datos.</li> <li>10. Si existen errores véase el flujo alterno 10.1</li> <li>11. Si los datos fueron introducidos correctamente, el sistema modifica los datos seleccionados terminando así el CASO DE USO.</li> </ol>
<b>Flujos Alternos</b>	
	10.1 El sistema retorna al paso 4.

### 2.6.5. Caso de uso Modificar Bloque

*Tabla 9 Descripción del Caso de uso Modificar Bloque*

Caso de uso	Modificar Bloque
<b>Objetivo</b>	Permite al administrador modificar un bloque.
<b>Actores</b>	Administrador(Inicia)
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción modificar de un bloque, luego muestra los campos con los datos, se hace la modificación y el sistema da la opción de guardar los cambios efectuados.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen y el bloque.
<b>Postcondiciones</b>	Bloque modificado en el sistema.
Flujo Normal de Eventos	
Acción de actor	Acción del sistema
1. Selecciona la opción Bloques de un volumen.	2. El sistema muestra un listado con todos los bloques disponibles del volumen.
3. Selecciona la opción modificar bloque.	4. El sistema muestra los campos con los datos para que estos puedan ser modificados.
5. Modifica los datos.	7. El sistema valida los datos.

6. Introduce los datos.	8. Si existen errores véase el flujo alternativo 8.1 9. Si los datos fueron introducidos correctamente, el sistema modifica los datos seleccionados terminando así el caso de uso.
<b>Flujos Alternos</b>	
	8.1 Muestra un mensaje de error y retorna al paso 4.

### 2.6.6. Caso de uso Mostrar Volumen

Tabla 10 Descripción del Caso de uso Mostrar Volumen

Caso de uso	Mostrar Volumen	
<b>Objetivo</b>	Permite al administrador mostrar un listado de volúmenes.	
<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción de Volúmenes después da en la opción Gestionar.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen.	
<b>Postcondiciones</b>	Existencia de volúmenes en el sistema.	
<b>Flujo Normal de Eventos</b>		
	<b>Acción de actor</b>	<b>Acción del sistema</b>
	1. Selecciona la opción Volúmenes.	2. Muestra la ayuda de los volúmenes y la opción Gestionar.
	3. Selecciona la opción Gestionar.	4. Muestra el listado con todos los volúmenes disponibles terminando así el caso de uso.

### 2.6.7. Caso de uso Mostrar Bloques

Tabla 11 Descripción del Caso de uso Mostrar Bloques

Caso de uso	Mostrar Bloques
<b>Objetivo</b>	Permite al administrador mostrar un listado de Bloques.

<b>Actores</b>	Administrador(Inicia)
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Bloques de un volumen del listado de volúmenes.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen.
<b>Postcondiciones</b>	Existencia de bloques en el sistema.
Flujo Normal de Eventos	
Acción de actor	Acción del sistema
1. Selecciona la opción Bloques del volumen	2. El sistema muestra un listado con todos los bloques disponibles del volumen terminando así el caso de uso.

### 2.6.8. Caso de uso Eliminar Volumen

Tabla 12 Descripción del Caso de uso Eliminar Volumen

<b>Caso de uso</b>	<b>Eliminar Volumen</b>		
<b>Objetivo</b>	Permite al administrador eliminar un volumen		
<b>Actores</b>	Administrador(Inicia)		
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción de Volúmenes después da en la opción Gestionar y el administrador dando clic en eliminar quedando eliminado el volumen.		
<b>Complejidad</b>	Media		
<b>Prioridad</b>	Crítico		
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen.		
<b>Postcondiciones</b>	Volumen eliminado del sistema.		
Flujo Normal de Eventos			
Acción de actor	Acción del sistema		
1. Selecciona la opción volúmenes.	2. Muestra la ayuda de los volúmenes y la opción Gestionar. 4. El sistema muestra el listado de volúmenes.		
3. Selecciona la opción	6. Elimina el volumen seleccionado, terminando así el caso de		

<p>Gestionar.</p> <p>5. Selecciona la opción eliminar volumen de un volumen del listado de volúmenes.</p>	<p>uso.</p>
---	-------------

### 2.6.9. Caso de uso Eliminar Bloque

Tabla 13 Descripción del Caso de uso Eliminar Bloque

Caso de uso	Eliminar Bloque	
<b>Objetivo</b>	Permite al administrador eliminar un bloque	
<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Bloques, se muestra un listado de bloques, se selecciona la opción eliminar, el sistema da la opción de guardar los cambios efectuados.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. Tiene que existir un volumen.	
<b>Postcondiciones</b>	Bloque eliminado del sistema.	
Flujo Normal de Eventos		
Acción de actor	Acción del sistema	
1. Selecciona la opción Bloques de un volumen.	2. El sistema muestra el listado de bloques del volumen.	
3. Selecciona la opción eliminar bloque, de un bloque de la lista.	4. Elimina el bloque seleccionado terminando así el caso de uso	

### 2.6.10. Caso de uso Administrar servicio de Volumen

Tabla 12 Descripción del Caso de uso Administrar servicio de Volumen

Caso de uso	Administrar servicio de Volumen
<b>Objetivo</b>	Permite al administrador administrar (iniciar, detener y reiniciar) un volumen.

<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción de Volúmenes después da en la opción Gestionar y el administrador dando clic la opción estado de un volumen se podrá iniciar o detener el mismo.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal.	
<b>Postcondiciones</b>	El servicio será iniciado o detenido.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Acción del sistema</b>	
<ol style="list-style-type: none"> <li>1. Selecciona la opción Volúmenes.</li> <li>2. Selecciona la opción Gestionar.</li> </ol>	<ol style="list-style-type: none"> <li>3. Muestra la ayuda de los volúmenes y la opción Gestionar.</li> <li>4. El sistema muestra un listado con todos los volúmenes disponibles.</li> </ol>	
<ol style="list-style-type: none"> <li>5. Selecciona una opción             <ol style="list-style-type: none"> <li>a) Si selecciona la opción iniciar volumen véase la sección Iniciar volumen.</li> <li>b) Si selecciona la opción detener volumen véase la sección Detener volumen.</li> </ol> </li> </ol>		
<b>Sección “Iniciar volumen”</b>		
<b>Acción del actor</b>	<b>Acción del sistema</b>	
<ol style="list-style-type: none"> <li>1. Selecciona la opción de iniciar.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema inicia el volumen seleccionado terminando así el caso de uso.</li> </ol>	
<b>Sección “Detener volumen”</b>		
<b>Acción del actor</b>	<b>Acción del sistema</b>	
<ol style="list-style-type: none"> <li>1. Selecciona la opción de detener.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema detiene el volumen seleccionado terminando así el caso de uso.</li> </ol>	

### 2.6.11. Caso de uso Administrar Servicio General

Tabla 13 Descripción del Caso de uso Administrar Servicio General

Caso de uso	Administrar Servicio General	
<b>Objetivo</b>	Permite al administrador administrar (iniciar, detener y reiniciar) el servicio de GlusterFSv3.2.	
<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Administrar Servicio. El sistema mostrara la ayuda y luego dará la posibilidad de iniciar, detener y reiniciar el servicio.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal.	
<b>Postcondiciones</b>	El servicio de GlusterFS será iniciado, reiniciado y modificado.	
Flujo Normal de Eventos		
Acción de actor	Acción del sistema	
1. Selecciona la opción Administrar Servicio.	2. Muestra la ayuda para la administración y la opción Administrar. 3. El sistema muestra las opciones de administración del servicio GlusterFSv3.2.	
4. Selecciona una opción. a) Si selecciona la opción iniciar servicio véase la sección Iniciar servicio. b) Si selecciona la opción reiniciar servicio véase la sección Reiniciar servicio. c) Si selecciona la opción detener servicio véase la sección Detener servicio.		
Sección "Iniciar servicio"		
Acción del actor	Acción del sistema	
1. Selecciona la opción de iniciar.	2. El sistema inicia el servicio de GlusterFSv3.2	

	terminando así el caso de uso.
<b>Sección “Reiniciar servicio”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. Selecciona la opción de reiniciar.	2. El sistema reinicia el servicio de GlusterFSv3.2 terminando así el caso de uso.
<b>Sección “Detener servicio”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
1. Selecciona la opción de detener.	2. El sistema detiene el servicio de GlusterFSv3.2 terminando así el caso de uso.

### 2.6.12. Caso de uso Administrar Plugin

*Tabla 15 Descripción del Caso de uso Administrar Plugin*

<b>Caso de uso</b>	<b>Administrar Plugin</b>	
<b>Objetivo</b>	Permite al administrador administrar el módulo de GlusterFS.	
<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Componentes de Zentyal, el sistema muestra tres pestañas con las opciones Instalar, Actualizar y Borrar dando la posibilidad de escoger la acción que se desee.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. El módulo GlusterFS debe existir en el repositorio.	
<b>Postcondiciones</b>	El módulo GlusterFS será instalado, actualizado o eliminado.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Acción del sistema</b>	
1. Selecciona la opción Gestión de software/Componentes de Zentyal.	2. Muestra la interfaz Componentes de Zentyal que contiene las opciones Instalar, Actualizar y Borrar.	
3. Selecciona una opción. a) Si selecciona la opción Instalar véase la sección		

<p>Instalar módulo.</p> <p>b) Si selecciona la opción Actualiza véase la sección Actualizar módulo.</p> <p>c) Si selecciona la opción Borrar véase la sección Borrar módulo.</p>	
<b>Sección “Instalar módulo”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
	<ol style="list-style-type: none"> <li>1. Muestra un listado con los módulos disponibles incluyendo el módulo GlusterFS.</li> </ol>
<ol style="list-style-type: none"> <li>2. Marca el checkbox correspondiente al módulo GlusterFS.</li> <li>3. Selecciona el botón Instalar.</li> </ol>	<ol style="list-style-type: none"> <li>4. Instala el módulo.</li> <li>5. Muestra un mensaje de verificación terminando así el caso de uso.</li> </ol>
<b>Sección “Actualizar módulo”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
	<ol style="list-style-type: none"> <li>1. Muestra un listado con los módulos que tengan actualizaciones.</li> </ol>
<ol style="list-style-type: none"> <li>2. Marca el checkbox correspondiente al módulo GlusterFS.</li> <li>3. Selecciona el botón Actualizar.</li> </ol>	<ol style="list-style-type: none"> <li>4. Actualiza el módulo.</li> <li>5. Muestra un mensaje de verificación terminando así el caso de uso.</li> </ol>
<b>Sección “Borrar módulo”</b>	
<b>Acción del actor</b>	<b>Acción del sistema</b>
	<ol style="list-style-type: none"> <li>1. Muestra un listado con los módulos instalados.</li> </ol>
<ol style="list-style-type: none"> <li>2. Marca el checkbox correspondiente al módulo GlusterFS.</li> <li>3. Selecciona el botón Borrar.</li> </ol>	<ol style="list-style-type: none"> <li>4. Borra el módulo.</li> <li>5. Muestra un mensaje de verificación terminando así el caso de uso.</li> </ol>

**2.6.13. Caso de uso Administrar Firewall**

Tabla 16 Descripción del Caso de uso Administrar Firewall

<b>Caso de uso</b>	<b>Instalar Firewall</b>	
<b>Objetivo</b>	Permite al administrador instalar firewall.	
<b>Actores</b>	Administrador(Inicia)	
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Componentes de Zentyal, el sistema muestra la opción Instalar permitiendo instalar el módulo Firewall.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. El módulo Firewall debe existir en el repositorio.	
<b>Postcondiciones</b>	El firewall será instalado	
<b>Flujo Normal de Eventos</b>		
	<b>Acción de actor</b>	<b>Acción del sistema</b>
	1. Selecciona la opción Gestión de software/Componentes de Zentyal.	2. Muestra la interfaz Componentes de Zentyal que contiene la opción Instalar.
	3. Selecciona la opción Instalar.	4. Muestra un listado con los módulos disponibles incluyendo el módulo Firewall.
	5. Marca el checkbox correspondiente al módulo Firewall. 6. Selecciona el botón Instalar.	7. Instala el módulo. 8. Muestra un mensaje de verificación terminando así el caso de uso.

**2.6.14. Caso de uso Administrar Firewall**

Tabla 17 Descripción del CASO DE USO Administrar Firewall

<b>Caso de uso</b>	<b>Instalar Antivirus</b>
<b>Objetivo</b>	Permite al administrador instalar el antivirus.

<b>Actores</b>	Administrador(Inicia)
<b>Resumen</b>	El caso de uso inicia cuando el administrador selecciona la opción Componentes de Zentyal, el sistema muestra la opción Instalar permitiendo instalar el módulo Antivirus.
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	El administrador debe estar autenticado en la plataforma Zentyal. El módulo Antivirus debe existir en el repositorio.
<b>Postcondiciones</b>	El antivirus será instalado
Flujo Normal de Eventos	
Acción de actor	Acción del sistema
1. Selecciona la opción Gestión de software/Componentes de Zentyal.	2. Muestra la interfaz Componentes de Zentyal que contiene la opción Instalar.
3. Selecciona la opción Instalar.	4. Muestra un listado con los módulos disponibles incluyendo el módulo Antivirus.
5. Marca el checkbox correspondiente al módulo Antivirus. 6. Selecciona el botón Instalar.	7. Instala el módulo. 8. Muestra un mensaje de verificación terminando así el caso de uso.

### 2.7. Diagrama de clases del diseño

Un diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. (Phelippe Kruchten 2000)

En la figura 3 se representan las ocho clases que harán posible la propuesta de solución. La clase GlusterFS es la clase principal la cual se encarga de realizar todas las operaciones del sistema; esta contiene una instancia de las clases ServicioG, Servidores y Volumes. Por su parte la clase ServicioG muestra las opciones administrativas del servicio de forma general. La clase Servidores muestra una vista

de todos los servidores introducidos en la aplicación y crea una instancia de la clase CurrentConnection; la misma muestra si el servidor está conectado o no. La clase Volumes muestra una vista de todos los volúmenes introducidos en la aplicación y además se relaciona con las clases Bloques, Setting y ServiciosV para mostrar a través de la clase Bloques los bloques pertenecientes al volumen, con la clase Setting las opciones de configuración del volumen y con la clase ServicioV da la posibilidad de administrar el volumen.

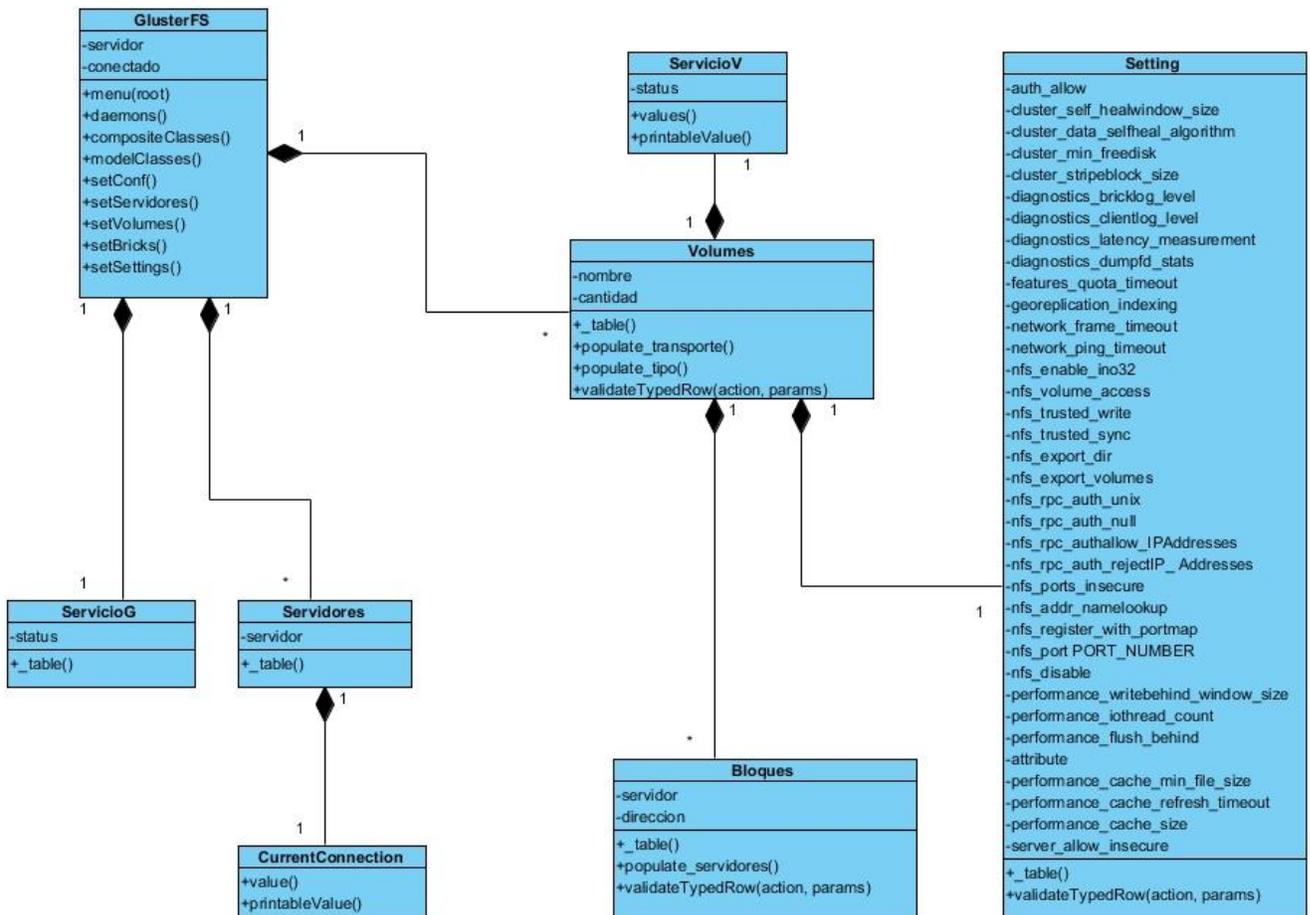


Figura 3 Diagrama de clases del diseño

### 2.8. Diagrama de clases del diseño con estereotipos web

El Diagrama de clases web no es más que una representación de clases y asociaciones. Los tipos de clases que se utilicen describirán la relación que tendrán al momento de la ejecución de la aplicación y cómo será su comunicación entre estas. (Phelippe Kruchten 2000)

En la figura 4 se modela el diagrama de clases con estereotipos web para el caso de uso Gestionar Servidores. La clase servidora server es la encargada de construir la clase cliente llamada Servidores, esta va a incluir las clases clientes Adicionar Servidores, Modificar Servidores y Eliminar Servidores. Tanto la clase Adicionar Servidores como Modificar Servidores implementan un formulario en el cual se introducirán los nuevos datos del servidor o modificaran los existentes respectivamente, que permitirá el envío de estos datos a la clase servidora server; la cual haciendo uso de la clase controladora GlusterFS, llevará a cabo las acciones pertinentes. La clase Eliminar Servidores realiza un redireccionamiento a la clase servidora server enviándole el identificador del servidor a eliminar, está haciendo uso de la clase controladora GlusterFS elimina el servidor.

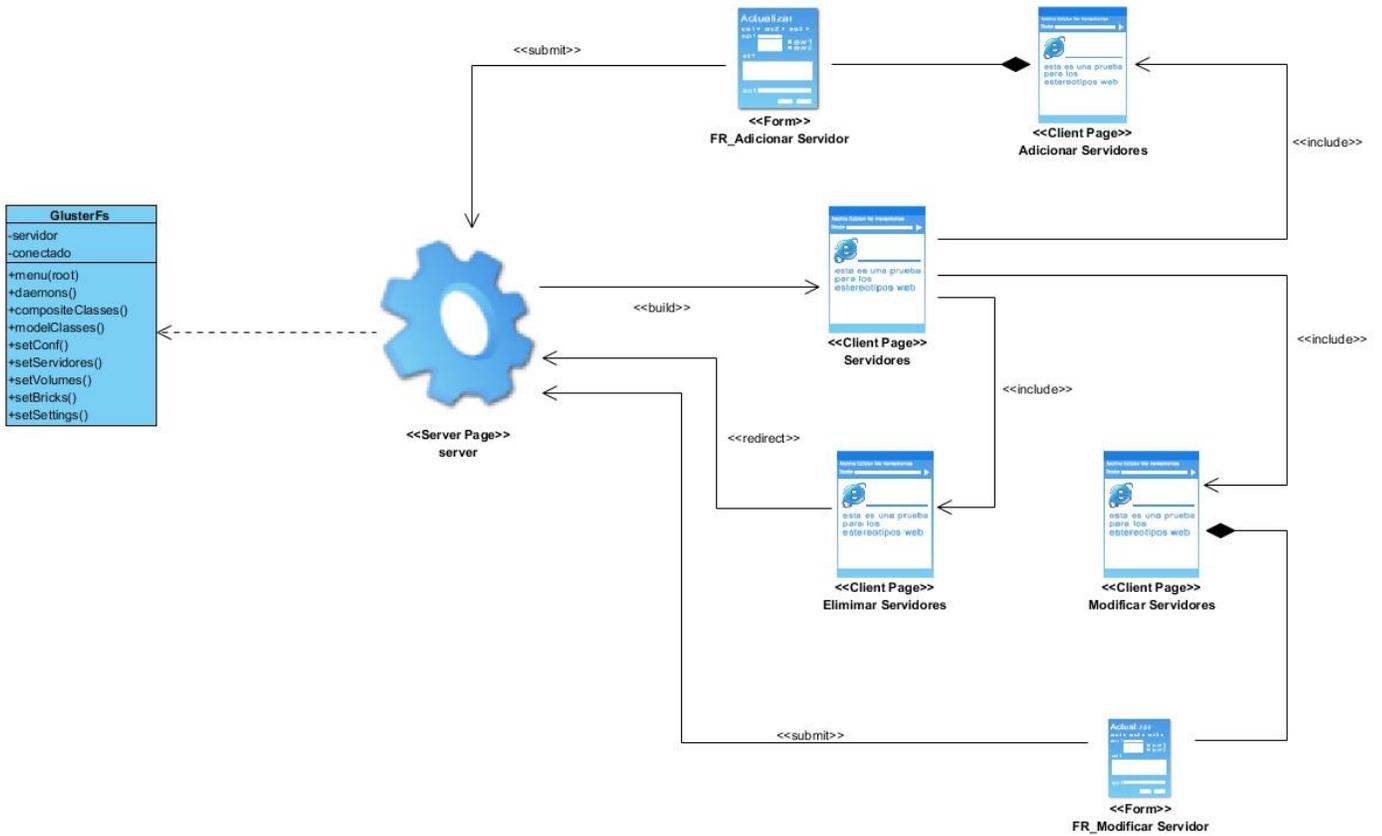


Figura 4 Diagrama de clases del diseño con estereotipos web para el caso de uso Gestionar Servidores

## 2.9. Diagrama de interacción

Los diagramas de interacción son diagramas que describen como los grupos de objetos colaboran para conseguir algún fin. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso. Los diagramas de interacción capturan el comportamiento de los casos de uso. Se expresan de dos formas: los diagramas de colaboración, los cuales muestran las relaciones entre los objetos y los mensajes que intercambian; y los diagramas de secuencia; los que describen las interacciones expresadas en función de secuencias temporales.

### 2.9.1. Diagrama de secuencia.

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados de interacción) se utilizan para la modelación de los aspectos dinámicos de un sistema. Un diagrama de este tipo muestra la

interacción de un conjunto de objetos y sus relaciones, teniendo como valor añadido este diagrama el hecho de mostrar los mensajes que se envían entre dichos objetos. (Roger S. Pressman 2001)

Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

La figura 5 refleja el diagrama de secuencia para el caso de uso Gestionar Servicio General. En este diagrama se distingue una línea de vida que pertenece a cada objeto y que representa su existencia a lo largo de un período de tiempo. La mayoría de estos objetos perdurarán en la medida que dure la interacción.

El administrador selecciona la opción Servicio General, el sistema obtiene el estado de servicio con el método estadoServicio() y se muestra en la vista el pid con el método PID(). El usuario tiene la opción de seleccionar Iniciar Servicio, Reiniciar Servicio o Detener Servicio. Si el usuario selecciona la opción Iniciar Servicio; el sistema a través del método addRow() inicia el demonio glusterfs-server. Si el usuario selecciona la opción Reiniciar Servicio; el sistema haciendo uso del método setRow () reinicia el demonio glusterfs-server. Si el usuario selecciona la opción Detener Servicio; el sistema mediante el método removeRow () detener el demonio glusterfs-server.

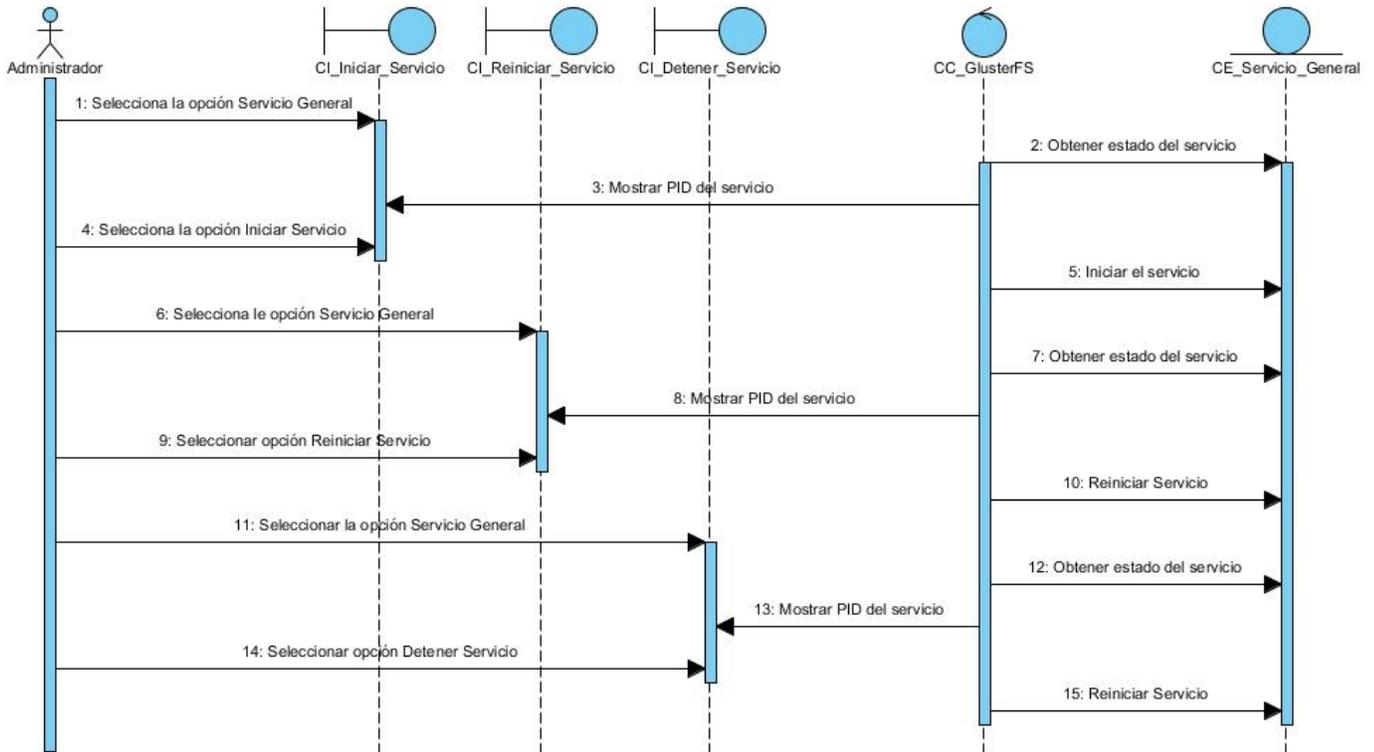


Figura 5 Diagrama de secuencia del CU Gestionar Servicio General

## 2.10. Arquitectura de software

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. (Roger S. Pressman 2001)

### 2.10.1. Modelo Vista Controlador

Para el desarrollo de la aplicación se hará uso del patrón de Arquitectura de software Modelo Vista Controlador (MVC), ya que Zentyal para la integración de los módulos genera una plantilla que implementa este modelo. MVC es un patrón que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. Este es muy utilizado en aplicaciones web. En nuestro caso la vista la va a proporcionar la interfaz gráfica de Zentyal junto con el código que provee los datos dinámicos a la misma. El modelo será el servicio que se quiere representar gráficamente o sea todas las clases del modelo que hacen posible la comunicación con el servicio de sistema de archivos

distribuidos; y el controlador es el responsable de recibir los eventos de entrada desde la vista facilitando los cambios en el modelo.

### **2.11. Patrones de diseño**

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (Jeffer Miguel Vargas 2012)

#### **2.11.1. Patrones GRASP**

GRASP es el acrónimo de General Responsibility Assignment Software Patterns. Una de las cosas más complicadas en Orientación a Objeto consiste en elegir las clases adecuadas y decidir cómo estas clases deben interactuar. (Jeffer Miguel Vargas 2012)

#### **2.11.2. Experto en información**

Este patrón consiste en asignar la responsabilidad a la clase que tiene la información necesaria para realizarla. En este caso la clase controladora GlusterFS es la que maneja toda la información de los datos brindados, por lo que las clases Bloques, Servidores, Volúmenes y Setting, que son las que se encargan de interactuar con el usuario, estas a su vez tienen la responsabilidad de validar los datos.

#### **2.11.3. Creador**

En la aplicación se pone de manifiesto este patrón cuando la clase controladora GlusterFS crea instancias de las clases Bloques, Servidores, Volúmenes y Setting para acceder a la información. Un ejemplo es el método `servidores_zentyal ()` de la clase GlusterFS el cual crea una instancia de la clase Servidores y mediante una estructura cíclica, utilizando los métodos `ids ()`, `row (id)` y `valueByName (parámetro)` obtiene una lista de los servidores de la aplicación.

### 2.11.4. Alta Cohesión

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas para que no realicen un trabajo enorme.

A menudo representan un alto grado de abstracción o han asumido responsabilidades que deberían haber delegado. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. No conviene este tipo de clases, porque: Son difíciles de comprender, reutilizar, conservar y son delicadas: las afectan constantemente los cambios.

Este patrón se evidencia en la clase GlusterFS de la aplicación, donde realiza todas las operaciones relacionadas con los datos sin afectarse entre ellas. Ejemplo de esto es: si se realiza la operación adicionar servidor, esta no afectaría otras operaciones tales como adicionar o eliminar volumen.

### 2.11.5. Bajo Acoplamiento

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de “muchas otras”. Una clase con alto (o fuerte) acoplamiento (depende de “muchas otras”) presenta los siguientes problemas: Los cambios de las clases afines ocasionan cambios locales. Son más difíciles de entender cuando están aisladas, de reutilizar porque se requiere la presencia de otras clases de las que dependen.

En la aplicación las clases de gestión de distintos tipos de datos, díganse Servidores y Volúmenes, tienen gran independencia entre ellas, por lo que la clase controladora puede hacer una mayor reutilización de las mismas, manifestándose así este patrón.

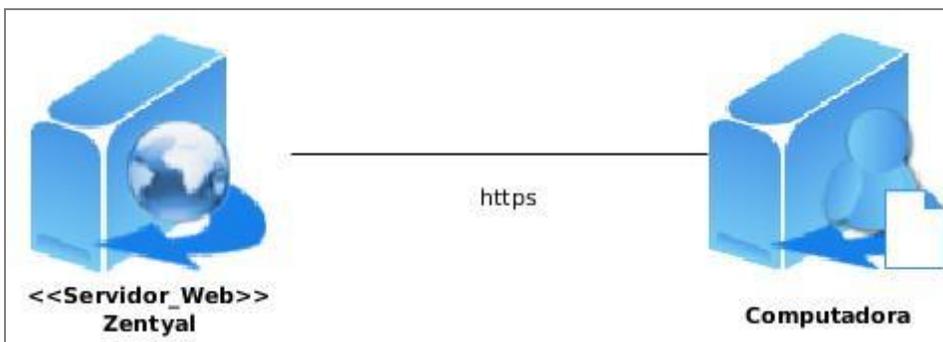
## 2.12. Diagrama de despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. (Ana Fernandez Vilas 2001)

A continuación se muestra el diagrama de despliegue en el cual se aprecia:

El nodo computadora: representa la máquina del administrador que hará uso de la prestación del servicio de sistema de archivos distribuidos a través del **Protocolo Seguro de Transferencia de Hipertexto (https)**.

El nodo Servidor Web: representa el servidor que notificará la disponibilidad del servicio de sistema de archivo distribuido a través de la solución de servidor Zentyal.



*Figura 6 Diagrama de despliegue del sistema*

### 2.13. Diagrama de componente

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre los componentes del software, sean éstos componentes de código fuente, binarios o ejecutables. Un paquete en un diagrama de componentes representa una división física del sistema. Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). (Pascal Alejandro 2002)

A continuación se muestra la figura 7 representando el diagrama de componentes para el módulo GlusterFS. Los componentes Bloques, Servidores, Volumes, ServicioG y Setting conforman el paquete Model el cual se encarga de definir el modelo de datos, este interactúa con el paquete CGI compuesto por los componentes Servidores, Volúmenes y ServicioG el cual representa la vista del módulo; el paquete Model también interactúa con el paquete Types que contiene los componentes Current y ServicioV conformando los tres, el paquete Zentyal que conjuntamente con el componente principal GlusterFS se

integran al paquete Sistema de archivos distribuidos, conformando el paquete NovaServer el cual representa el módulo a desarrollar.

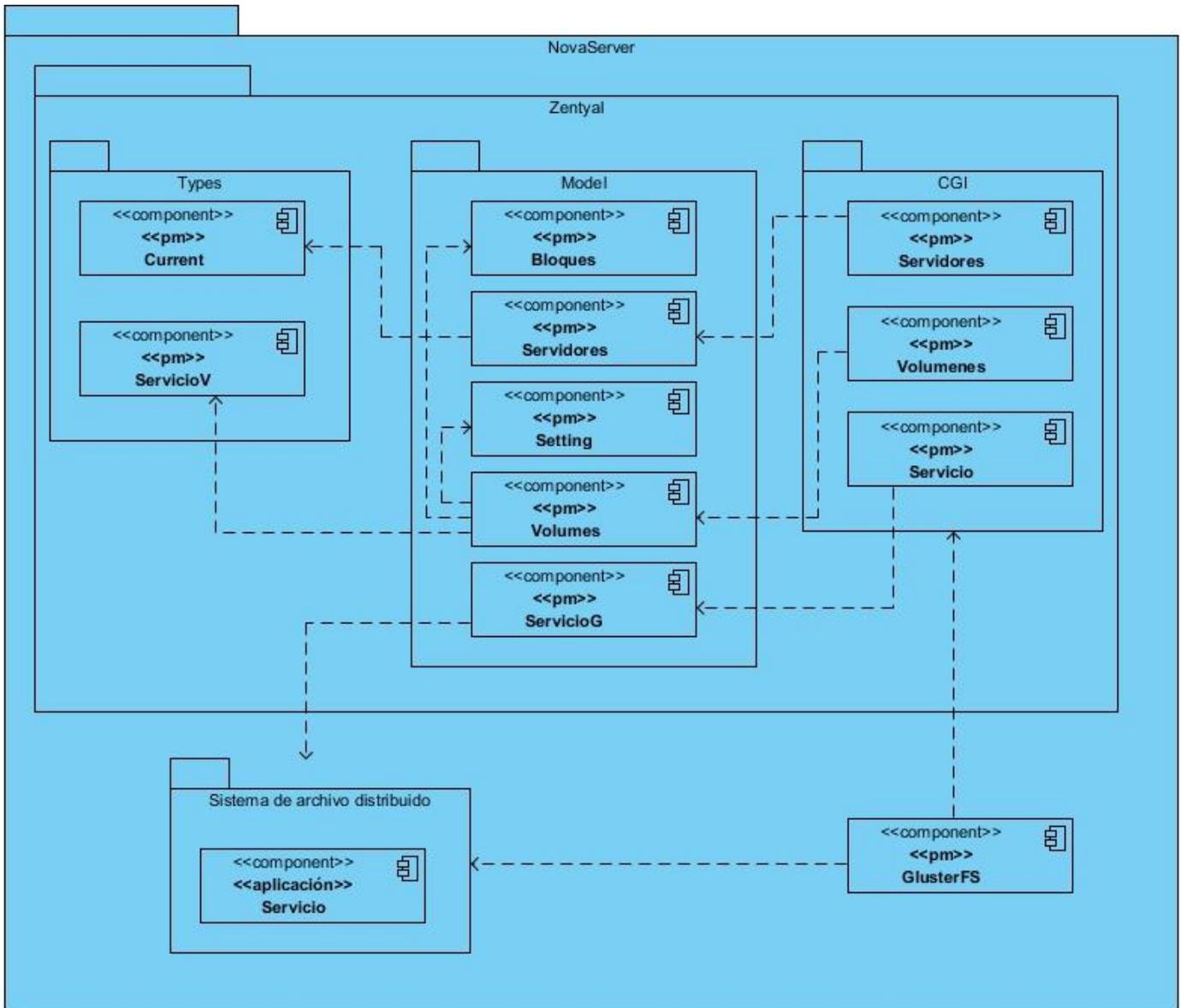


Figura 7 Diagrama de componentes del sistema

### 2.14. Conclusiones del capítulo

A lo largo de este capítulo se realizó el análisis y diseño de la aplicación. Se definieron los requisitos funcionales y no funcionales para el funcionamiento de la misma. Se especificaron 14 casos de uso permitiendo modelar el Diagrama de Casos de uso del sistema, el Diagrama de Clases del diseño y los Diagramas de secuencia para una mejor comprensión de las funcionalidades del sistema. Se hizo uso del patrón arquitectónico Modelo Vista Controlador permitiendo crear una arquitectura robusta, empleando las mejores prácticas de diseño. Con la realización del módulo para la plataforma Zentyal el administrador podrá prestar el servicio de GlusterFSv3.2 de forma cómoda e intuitiva desde un ordenador.

## Capítulo 3. Implementación y prueba del sistema.

### 3.1. Introducción

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados. Los resultados son observados y registrados. Una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. En este capítulo se llevaran a cabo pruebas de caja negra en una segunda iteración.

### 3.2. Pruebas de caja negra

Las pruebas de caja negra son las que se realizan en la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo, fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

A continuación se presentan los casos de pruebas para cada uno de los casos de uso planteados a través de la matriz de datos, permitiendo comprobar que la aplicación funcione de forma correcta, donde:

- V: indica válido
- I: indica inválido
- NA: indica que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Además se muestra la matriz parcial por cada caso de uso.

## Capítulo 3. Implementación y prueba del sistema

Tabla 14 Matriz parcial del escenario para el Caso de uso Gestionar Servidores.

Nombre del Escenario	Flujo donde empieza
Escenario 1-Introducir un nuevo servidor	Flujo Básico
Escenario 2-No se puede introducir un nuevo servidor conflicto en la dirección.	Flujo Básico
Escenario 3- No se puede introducir un nuevo servidor, nombre incorrecto.	Flujo Básico
Escenario 4-Modificar un servidor.	Flujo Básico
Escenario 5-No se puede modificar el nombre, conflictos de repetición.	Flujo Básico
Escenario 6-No se puede modificar la dirección	Flujo Básico
Escenario 7-Eliminar un servidor	Flujo Básico

Tabla 15 Matriz de caso de prueba para el Caso de uso Gestionar Servidores.

ID del caso de prueba	Escenario/Condición	Dirección del servidor	Resultado esperado
RC1	Escenario1-Introducir un nuevo servidor	V	Visualiza un nuevo servidor en la lista de servidores.
RC2	Escenario2-No se puede introducir un nuevo servidor conflicto en la dirección.	I	Muestra un mensaje de error, vuelta al paso 1
RC3	Escenario3- No se puede introducir un nuevo servidor, ausencia de la dirección.	N/A	Muestra un mensaje de error, vuelta al paso 1.
RC4	Escenario7- Eliminar un servidor	N/A	Elimina el servidor seleccionado.

## Capítulo 3. Implementación y prueba del sistema

Tabla 17 Matriz de caso de prueba con los valores para los datos.

ID del caso de prueba	Escenario/Condición	Dirección del servidor	Resultado esperado
RC1	Escenario1-Introducir un nuevo servidor	10.8.122.102	Nuevo servidor en el sistema.
RC2	Escenario2-No se puede introducir un nuevo servidor conflicto en la dirección.	Se introduce una dirección existente:10.8.122.101	Mensaje de error: El servidor 10.8.122.101 ya existe.
RC3	Escenario3- No se puede introducir un nuevo servidor, ausencia de la dirección.	N/A	Mensaje de error: El servidor necesita argumentos.
RC4	Escenario7- Eliminar un servidor	N/A	Servidor eliminado del sistema.

Tabla 18 Matriz parcial del escenario para el Caso de uso Insertar Volumen.

Nombre del Escenario	Flujo donde empieza
Escenario1-Introducir un volumen	Flujo Básico
Escenario2-Conflicto con el nombre	Flujo Básico
Escenario3-Cantidad no precisada	Flujo Básico
Escenario4-No existen bloques	Flujo Básico

Tabla 169 Matriz de caso de prueba para el Caso de uso Insertar Volumen.

ID del caso de prueba	Escenario/Condición	Nombre del volumen	Cantidad	Bloques	Resultado esperado
RC1	Escenario1-Introducir un	V	V	V	Muestra el nuevo volumen en la

## Capítulo 3. Implementación y prueba del sistema

	volumen				lista de volúmenes
RC2	Escenario2-Conflicto con el nombre	I	V	N/A	Muestra un mensaje de error, retorna al paso 4.
RC3	Escenario3-Cantidad no precisada	V	I	N/A	Muestra un mensaje de error, retorna al paso 4.
RC4	Escenario4-No existen Bloques	V	V	I	No se adiciona el volumen

Tabla 20 Matriz de caso de prueba con los valores para los datos.

ID del caso de prueba	Escenario/Condición	Nombre del volumen	Cantidad	Bloques	Resultado esperado
RC1	Escenario1-Introducir un volumen	Volumen1	0	Se adicionan 1 o más bloques.	Muestra el nuevo volumen en la lista de volúmenes.
RC2	Escenario2-Conflicto con el nombre	****	0	N/A	Muestra un mensaje especificando que el nombre debe ser alfanumérico.
RC3	Escenario3-Cantidad no precisada	nuevo	0	N/A	Mensaje de error: La cantidad de stripe o replica debe ser mayor

## Capítulo 3. Implementación y prueba del sistema

					que 2.
RC4	Escenario4-No existen Bloques	Nuevo1	0	No se adiciona ningún bloque	Mensaje de error: Inserte al menos un bloque.

Tabla 21 Matriz parcial del escenario para el Caso de uso Insertar Bloque.

Nombre del Escenario	Flujo donde empieza
Escenario1-Introducir un bloque	Flujo Básico
Escenario2-Conflicto con la dirección	Flujo Básico
Escenario3-Conflicto con el servidor	Flujo Básico

Tabla 22 Matriz de caso de prueba para el Caso de uso Insertar Bloque.

ID del caso de prueba	Escenario/Condición	Servidor	Dirección	Resultado esperado
RC1	Escenario1-Introducir un bloque	V	V	Adiciona el bloque
RC2	Escenario2-Conflicto con la dirección	V	I	Muestra mensaje de error, retorna al paso 4
RC3	Escenario3-Conflicto con el servidor	I	V	Muestra mensaje de error, retorna al paso 4

Tabla 23 Matriz de caso de prueba con los valores para los datos.

ID del caso de prueba	Escenario/Condición	Servidor	Dirección	Resultado esperado
RC1	Escenario1-Introducir un bloque	10.8.122.101	/data	Adiciona el bloque a la lista de bloques
RC2	Escenario2-	10.8.122.101	123	Muestra un mensaje de

## Capítulo 3. Implementación y prueba del sistema

	Conflicto con la dirección			error:
RC3	Escenario3-Conflicto con el servidor	No se selecciona ningún servidor	/data1	Mensaje de error: Se necesita argumentos del servidor.

Tabla 24 Matriz parcial del escenario para el Caso de uso Modificar Volumen.

Nombre del Escenario	Flujo donde empieza
Escenario1-Modificar volumen	Flujo Básico
Escenario2-Conflicto con el nombre	Flujo Básico
Escenario3-Conflicto con el tipo y la cantidad	Flujo Básico

Tabla 25 Matriz de caso de prueba para el Caso de uso Modificar Volumen.

ID del caso de prueba	Escenario/Condición	Nombre	Tipo	Cantidad	Resultado
RC1	Escenario1-Modificar volumen	V	N/A	V	Se modifica el volumen
RC2	Escenario2-Conflicto con el nombre	I	N/A	N/A	Muestra un error, retorna al paso 4

Tabla 26 Matriz de caso de prueba con los valores para los datos.

ID del caso de prueba	Escenario/Condición	Nombre	Tipo	Cantidad	Resultado
RC1	Escenario1-Modificar volumen	Volumen vol	N/A	34	Se modifica el volumen en la lista de volúmenes
RC2	Escenario2-	Nuevo"	N/A	N/A	Mensaje de error:

## Capítulo 3. Implementación y prueba del sistema

	Conflicto con el nombre				El nombre debe ser alfanumérico.
--	-------------------------	--	--	--	----------------------------------

Tabla 27 Matriz parcial del escenario para el Caso de uso Modificar Bloque.

Nombre del Escenario	Flujo donde empieza
Escenario1-Modificar bloque	Flujo Básico
Escenario2-Conflicto con el servidor	Flujo Básico
Escenario3-Conflicto con la dirección.	Flujo Básico

Tabla 28 Matriz de caso de prueba para el Caso de uso Modificar Bloque.

ID del caso de prueba	Escenario/Condición	Servidor	Dirección	Resultado esperado
RC1	Escenario1- Modificar bloque	V	V	Bloque modificado

Tabla 29 Matriz de caso de prueba con los valores para los datos.

ID del caso de prueba	Escenario/Condición	Servidor	Dirección	Resultado esperado
RC1	Escenario1- Modificar bloque	10.8.122.101	/data5	Bloque modificado en la lista de bloques.

### 3.3. Conclusiones del capítulo

En el presente capítulo se concibió un conjunto pruebas para ser aplicadas al módulo, con el objetivo de medir la calidad del producto realizado y validar la solución propuesta. Se aplicaron un total de 20 diseño de casos de pruebas, de los cuales se identificaron en una segunda iteración 2 no conformidades las mismas no procedieron, por lo tanto se concluye con la obtención de un producto de software listo para ser utilizado y desplegado.

### Conclusiones

Al concluir el presente trabajo:

- Se realizó un estudio del estado del arte de diferentes sistemas de archivos distribuidos a partir del cual se decidió utilizar al GlusterFS para la implantación del sistema de archivos distribuidos permitiendo aprovechar grandes características distintivas.
- Se llevó a cabo la implementación del módulo de sistema de archivos distribuidos para integrárselo a la plataforma Zentyal obteniéndose la prestación del servicio de sistema de archivos distribuidos GlusterFS de forma intuitiva y estable.
- Se validó la solución propuesta mediante casos de prueba de caja negra, obteniéndose resultados satisfactorios y cumpliendo así los objetivos propuestos.

El resultado de esta investigación no solo va a posibilitar que los administradores del país implementen con computadoras comunes sistemas de almacenamiento distribuido, sino que además utilizando la documentación generada en esta investigación, los futuros desarrolladores de Nova podrán implementar nuevos módulos o modificar los existentes maximizando los niveles de Seguridad, Sostenibilidad, Soberanía y Socio – Adaptabilidad de la distribución. Haciendo a Nova para servidores el sistema operativo idóneo para los administradores de red del país.

### Recomendaciones

Se recomienda:

- La implementación de una funcionalidad que permita administrar el balance de volúmenes a través de la interfaz implementada
- La implementación de una funcionalidad que permita administrar la gestión de geo-replicación a través de la interfaz implementada.
- La implementación de una funcionalidad que permita administrar las listas de control de acceso con POSIX a través de la interfaz implementada.

### Referencias bibliográficas

ECURED, 2009. OpenUp - EcuRed. In: [online]. 2009. [Accessed 4 junio 2012]. Available from: <http://www.ecured.cu/index.php/OpenUp>.

F. J. MORA, 2008. Administración de servicios de Internet: De la teoría a la práctica - D Marcos Jorquera - Google Libros. In: [online]. 2008. [Accessed 4 junio 2012]. Available from: [http://books.google.com.cu/books?id=PnYgzkE7XdYC&pg=PA288&lpg=PA288&dq=celda+coda&source=bl&ots=P7kRgOFJF-&sig=RIZhus3migwC05polqxgtl-Inj4&hl=es&sa=X&ei=H\\_zMT8ihLli\\_0AH33P2sDw&ved=0CEwQ6AEwAQ#v=onepage&q=autor&f=false](http://books.google.com.cu/books?id=PnYgzkE7XdYC&pg=PA288&lpg=PA288&dq=celda+coda&source=bl&ots=P7kRgOFJF-&sig=RIZhus3migwC05polqxgtl-Inj4&hl=es&sa=X&ei=H_zMT8ihLli_0AH33P2sDw&ved=0CEwQ6AEwAQ#v=onepage&q=autor&f=false).

KELLY LOPEZ, 2012. *Archivos Distribuidos- Presentation Transcript* [online]. S.l.: s.n. Available from: <http://www.slideshare.net/bonnzai/archivos-distribuidos-presentation>.

LUIS ALBERTO CUENCA, 2011. Tesis. In: [online]. 2011. [Accessed 4 junio 2012]. Available from: <http://www.docstoc.com/docs/113763395/Tesis>.

UNIVERSIDAD NUEVA ESPARTA, 2011. Marco Teórico. In: [online]. 2011. [Accessed 4 junio 2012]. Available from: <http://sod-archivos.tripod.com/marco.htm>. -

MAIKEL RAMÍREZ, MILLÁN, Eduardo y Valery Moreno, 2011. *Herramienta para programar un controlador lógico programable basado en hardware reconfigurable* [online]. S.l.: s.n. Available from: [http://rielac.cujae.edu.cu/index.php/rieac/article/view/83/pdf\\_77](http://rielac.cujae.edu.cu/index.php/rieac/article/view/83/pdf_77).

ROGER S. PRESSMAN, 2001. Ingeniería de Software, un enfoque práctico. S.l.: s.n.

PASCAL ALEJANDRO, 2002. Manual de UML. In: [online]. 2002. [Accessed 7 junio 2012]. Available from: <http://es.scribd.com/doc/80174942/Manual-de-UML>.

PHELIPPE KRUCHTEN, 2000. The Rational Unified Process: An Introduction. 2da edición. S.l.: s.n.

JEFFER MIGUEL VARGAS, 2012. Conceptos de Ingeniería del Software. In: [online]. 2012. [Accessed 7 junio 2012]. Available from: <http://jeffervargas.blogspot.com/>.

ANA FERNANDEZ VILAS, 2001. Diagramas de Despliegue. In: [online]. 2001. [Accessed 7 junio 2012]. Available from: <http://es.scribd.com/doc/91245075/Diagramas-de-Despliegue>.

IVAR JACOBSON, 2000. El proceso unificado de desarrollo de software. S.l.: Félix Varela.

## Bibliografía

GLUSTER, 2011. Gluster File System - 3.2 Administration Guide. S.l.: s.n.

Documentación de Zentyal 2.2. 2011. Primeros pasos con Zentyal — Documentación de Zentyal 2.2. In: [online]. 2011. [Accessed 4 junio 2012]. Available from: <http://doc.zentyal.org/es/firststeps.html#configuracion-de-red-en-zentyal>.

Documentación de Zentyal 2.2, 2011. Documentation/Community/Document/Development/Tutorial/Version0 – Zentyal. In: [online]. 2011. [Accessed 4 junio 2012]. Available from: <http://trac.zentyal.org/wiki/Documentation/Community/Document/Development/Tutorial/Version0>.

KELLY LOPEZ, 2012. *Archivos Distribuidos- Presentation Transcript* [online]. S.l.: s.n. Available from: <http://www.slideshare.net/bonnzai/archivos-distribuidos-presentation>.

LUIS ALBERTO CUENCA, 2011. Tesis. In: [online]. 2011. [Accessed 4 junio 2012]. Available from: <http://www.docstoc.com/docs/113763395/Tesis>.

UNIVERSIDAD NUEVA ESPARTA, 2011. Marco Teórico. In: [online]. 2011. [Accessed 4 junio 2012]. Available from: <http://sod-archivos.tripod.com/marco.htm>. -

MAIKEL RAMÍREZ, MILLÁN, Eduardo y Valery Moreno, 2011. *Herramienta para programar un controlador lógico programable basado en hardware reconfigurable* [online]. S.l.: s.n. Available from: [http://rielac.cujae.edu.cu/index.php/rieac/article/view/83/pdf\\_77](http://rielac.cujae.edu.cu/index.php/rieac/article/view/83/pdf_77).

ROGER S. PRESSMAN, 2001. Ingeniería de Software, un enfoque práctico. S.l.: s.n.

PASCAL ALEJANDRO, 2002. Manual de UML. In: [online]. 2002. [Accessed 7 junio 2012]. Available from: <http://es.scribd.com/doc/80174942/Manual-de-UML>. -

PHELIPPE KRUCHTEN, 2000. The Rational Unified Process: An Introduction. 2da edición. S.l.: s.n.

JEFFER MIGUEL VARGAS, 2012. Conceptos de Ingeniería del Software. In: [online]. 2012.

IVAR JACOBSON, 2000. El proceso unificado de desarrollo de software. S.l.: Félix Varela.

### Glosario de términos

**SAD:** Sistema de archivos distribuidos.

**AFS2:** Sistema de archivos distribuidos Andrews File System.

**RVM:** Memoria Virtual Recuperable.

**Clog:** Programa para identificarse ante el servidor de identificación.

**venus:** Es un demonio que se ejecuta únicamente en los clientes CODA.

**RPC:** (del inglés Remote Procedure Call, Llamada a Procedimiento Remoto) es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

**CML:** Registro de modificaciones.

**RG:** Grupos de recurso.

**SCSI:** reserve or release (Liberación o reserva de SCSI): es una protección en el nivel del dispositivo que es usada por Enterprise Media Manager (EMM) para evitar compartir involuntariamente dispositivos de cinta y tener problemas posibles de pérdida de datos.

**Superblock:** Se utiliza para guardar datos y almacenar los metadatos que referencien a dichos datos.

**VFS:** Sistema de archivo virtual.

**UNIX:** Designa el núcleo de un sistema operativo multiusuario y multitarea.

**NFS:** (del inglés Network File System, Sistema de archivos de red), es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuidos en un entorno de red de computadoras de área local.

**I/O:** (Entrada y salida), hace referencia a la entrada y salida de la información en un medio.

**RDMA:** Remote Direct Memory Access permite mover datos directamente desde la memoria de un computador a otro. Esto permite alto rendimiento y baja latencia de red.

**Peta-bytes:** Un peta-byte es una unidad de almacenamiento de información cuyo símbolo es el PB, y equivale a 10<sup>15</sup>bytes.

**LAN:** (red local o del inglés Local Area Network) es la interconexión de una o varias computadoras y periféricos.

**WAN:** (del inglés wide area network, red de área amplia) es un tipo de red de computadoras capaz de cubrir distancias desde unos 100 hasta unos 1000 km, proporcionando un servicio a un país o un continente.

**API:** (del inglés Application Programming Interface, Interfaz de programación de aplicaciones) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas vulgarmente "librerías").

**Fuse:** Módulo para el kernel, ese es un soporte para GlusterFS.

**XFS:** Es un sistema de archivos de 64 bits con journaling de alto rendimiento creado por SGI (antiguamente Silicon Graphics Inc.) para su implementación de UNIX llamada IRIX.

**FOSS:** Software libre y de código abierto (también conocido como FLOSS, siglas de free/libre and open source software, en inglés) es el software que está licenciado de tal manera que los usuarios pueden estudiar, modificar y mejorar su diseño mediante la disponibilidad de su código fuente.

**PYMEs:** Pequeñas y medianas empresas con características distintivas, y tiene dimensiones con ciertos límites ocupacionales y financieros prefijados por los Estados o regiones.

**MVC:** Modelo Vista Controlador, es un patrón de arquitectura de software.

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

**CASE:** Computer Aided Software Engineering, en español Ingeniería de Software Asistida por ordenador.

**UML:** Unified Modeling Language, en español Lenguaje Unificado de Modelado.