

**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**ESTRATEGIA DE IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA  
DISPONIBILIDAD DE N NODOS SOBRE LINUX USANDO SOFTWARE  
LIBRE.**

**GERSON CÁCERES**

**TESIS DE GRADO PRESENTADA COMO REQUISITO  
PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA DE SISTEMAS**

**Quito, Agosto 2012**

**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**COLEGIO POLITÉCNICO**

**HOJA DE APROBACION DE TESIS**

**ESTRATEGIA DE IMPLEMENTACIÓN DE UN CLÚSTER DE ALTA  
DISPONIBILIDAD DE N NODOS SOBRE LINUX USANDO SOFTWARE  
LIBRE.**

**GERSON CÁCERES**

Fausto Pasmay, M.Sc.

Director de Tesis .....  
.....

Luis Miguel Prócel, M.Sc.

Miembro del Comité de Tesis .....  
.....

Santiago Gangotena, Ph.D.

Decano del Colegio Politécnico .....  
.....

Quito, Agosto 2012

© Derechos de autor  
**Gerson Cáceres**  
2012

## **RESUMEN**

Un clúster de alta disponibilidad permite el acceso continuo a los servicios y aplicaciones más dentro de una organización. El siguiente trabajo presenta una estrategia para la implementación de un clúster de alta disponibilidad de N nodos sobre Linux usando software libre. Este documento menciona los componentes más importantes del clúster y expone los conceptos más importantes. También explica la función que cumple cada uno de los componentes y como deben ser integrados dentro del clúster. Por último muestra como instalar, configurar y probar un clúster de alta disponibilidad con tres nodos para los servicios de transferencia de archivos, resolución de nombres, y servidor de páginas Web.

**ABSTRACT**

A high availability cluster allows continued access to services and applications within an organization. The work below presents a strategy for implementing a high availability cluster of N nodes on Linux using free software. This document lists all major components of the cluster and presents the most important concepts, it explains the role of each of the components and how they are integrated into the cluster. Finally, the document shows how to install, configure and test a high availability cluster with three nodes for file transfer services, name resolution, and Web page server.

## TABLA DE CONTENIDO

	Pág.
CAPITULO I .....	1
1. INTRODUCCION .....	1
1.1. GENERALIDADES.....	1
1.2. ANTECEDENTES.....	1
1.3. OBJETIVOS.....	4
1.3.1. OBJETIVO FINAL .....	4
1.3.2. OBJETIVOS ESPECIFICOS .....	5
1.4. JUSTIFICACION.....	5
1.5. ALCANCE .....	6
1.6. ASPECTOS METODOLOGICOS.....	6
1.6.1. IDENTIFICACIÓN Y LOCALIZACIÓN DEL SOFTWARE .....	7
1.6.2. EVALUACIÓN Y SELECCIÓN DEL SOFTWARE .....	8
1.6.3. OBTENCIÓN DEL SOFTWARE Y SU DOCUMENTACIÓN .....	8
1.6.4. INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE .....	9
1.7. REQUERIMIENTOS BÁSICOS.....	9
1.8. RENUNCIA DE RESPONSABILIDAD.....	9
CAPITULO II .....	11
2. DEFINICIONES Y CONCEPTOS IMPORTANTES .....	11
2.1. CLÚSTER.....	11
2.1.1. TIPOS DE CLUSTERS .....	13
2.1.2. COMPONENTES PRINCIPALES DE UN CLÚSTER DE ALTA DISPONIBILIDAD .....	15
2.1.3. MODELOS DE IMPLEMENTACIÓN .....	19
2.1.4. ARQUITECTURA .....	20
2.1.5. SISTEMA DE ARCHIVOS PARA CLÚSTERES .....	21
2.1.6. OTRAS DEFINICIONES .....	22
2.1.7. BENEFICIOS DE UN CLÚSTER DE ALTA DISPONIBILIDAD .....	26
2.2. SISTEMA OPERATIVO.....	27
2.2.1. SISTEMA OPERATIVO GNU/LINUX .....	27

2.3. SERVIDOR WEB.....	29
2.4. SERVIDOR DE NOMBRES DE DOMINIO.....	30
2.5. SERVIDOR DE TRANSFERENCIA DE ARCHIVOS.....	30
CAPITULO III .....	32
3. ESTRATEGIA DE IMPLEMENTACION .....	32
3.1. DISEÑO DEL CLÚSTER.....	33
3.2. INSTALACIÓN DEL SISTEMA OPERATIVO.....	37
3.2.1. ACTUALIZACIONES DEL SISTEMA OPERATIVO .....	37
3.2.2. RED LAN DE COMUNICACIÓN TCP/IP .....	39
3.3. INSTALACIÓN DEL SISTEMA DE ARCHIVOS PARA CLÚSTERES Ó DEL SISTEMA DE ALMACENAMIENTO DISTRIBUIDO .....	39
3.3.1. SISTEMA DE ARCHIVOS PARA CLÚSTERES. ....	40
3.3.2. SISTEMA DE ALMACENAMIENTO DISTRIBUIDO. ....	41
3.4. INSTALACIÓN DEL ADMINISTRADOR DE MEMBRECÍA Y MENSAJERÍA. ....	42
3.5. INSTALACIÓN DEL ADMINISTRADOR DE RECURSOS.....	44
3.6. INSTALACIÓN DE LOS SERVICIOS y APLICACIONES.....	45
CAPITULO IV .....	46
4. IMPLEMENTACION DEL CLÚSTER .....	46
4.1. DISEÑO DEL CLÚSTER CON 2 NODOS.....	46
4.2. REQUERIMIENTOS DE HARDWARE.....	48
4.3. REQUERIMIENTOS DE SOFTWARE.....	48
4.4. INSTALACION Y CONFIGURACIÓN DEL SISTEMA OPERATIVO CentOS .....	49
4.5. PREPARATIVOS.....	51
4.6. CONFIGURACIONES ADICIONALES.....	52
4.7. INSTALACIÓN DEL SISTEMA DE ALMACENAMIENTO DISTRIBUIDO - DRBD .....	53
4.7.1. VISIÓN GENERAL DE DRBD .....	54
4.7.2. ROLES .....	56
4.7.3. CONVENCIONES GENERALES DE DRBD .....	57
4.7.4. INSTALACIÓN DE DRBD .....	58

4.7.5. CONFIGURACIÓN INICIAL DE DRBD .....	58
4.8. INSTALACIÓN DEL ADMINISTRADOR DE MEMBRECÍA Y MENSAJERÍA - COROSYNC .....	58
4.8.1. COMPONENTES PRINCIPALES DE COROSYNC .....	60
4.8.1.1. GRUPO DE PROCESO CERRADO - CPG .....	60
4.8.1.2. QUORUM .....	61
4.8.1.3. SINCRONÍA VIRTUAL EXTENDIDA - DEVS .....	61
4.8.1.4. CONFDB .....	61
4.8.1.5. CFG .....	61
4.9. INSTALACIÓN DEL ADMINISTRADOR DE RECURSOS-PACEMAKER	62
4.9.1. VISIÓN GENERAL DE PACEMAKER .....	63
4.9.1. COMPONENTES INTERNOS DE PACEMAKER .....	63
4.9.2. AGENTES .....	65
4.9.3. TIPO DE RECURSOS .....	66
4.9.4. RESTRICCIONES DE LOS RECURSOS .....	67
4.9.5. PUNTUACIONES E INFINITO .....	68
4.9.6. NODOS DE RECUPERACIÓN .....	69
4.10. INSTALACIÓN DE LOS RECURSOS .....	70
4.10.1. INSTALACIÓN DEL SERVIDOR PARA RESOLUCIÓN DE NOMBRES DE DOMINIO .....	70
4.10.2. INSTALACIÓN DEL SERVIDOR DE TRANSFERENCIA DE ARCHIVOS .....	71
4.11. AGREGAR UN TERCER NODO AL CLÚSTER ACTUAL .....	72
4.11.1. DISEÑO DEL CLÚSTER CON 3 NODOS .....	72
4.11.2. CONFIGURACIÓN DEL CLÚSTER CON 3 NODOS .....	74
4.11.3. INSTALACIÓN DEL SERVIDOR WEB .....	75
4.12. ESTRATEGIA PARA AGREGAR NODOS ADICIONALES AL CLÚSTER HA .....	75
4.13. COMANDOS PRINCIPALES PARA LA ADMINISTRACIÓN DEL CLÚSTER .....	77
CAPITULO V .....	78
5. PRUEBAS FUNCIONALES Y RESULTADOS .....	78

5.1.	PRUEBAS DE REPLICACIÓN DEL CLÚSTER CON 2 NODOS.....	78
5.2.	PRUEBAS DE MIGRACIÓN DEL CLÚSTER CON 2 NODOS.....	81
5.3.	PRUEBAS FUNCIONALES DEL CLÚSTER CON 3 NODOS.....	82
5.4.	PRUEBAS DE MIGRACIÓN DEL CLÚSTER CON 3 NODOS.....	83
6.	CONCLUSIONES Y RECOMENDACIONES .....	85
6.1.	CONCLUSIONES .....	85
6.2.	RECOMENDACIONES .....	89
	BIBLIOGRAFÍA .....	93
	GLOSARIO DE TÉRMINOS .....	96

## **ANEXOS**

ANEXO 1	INSTALACIÓN Y CONFIGURACIÓN DEL CENTOS .....	102
ANEXO 2	INSTALACIÓN y CONFIGURACIÓN DE DRBD .....	123
ANEXO 3	INSTALACIÓN Y CONFIGURACIÓN DE COROSYNC .....	129
ANEXO 4	INSTALACIÓN Y CONFIGURACIÓN DE PACEMAKER .....	134
ANEXO 5	INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DNS .....	136
ANEXO 6	INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR FTP .....	143
ANEXO 7	CONFIGURACIÓN DE LOS RECURSOS .....	144
ANEXO 8	CONFIGURACIÓN DEL TERCER NODO .....	147
ANEXO 9	INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR HTTP .....	153
ANEXO 10	PRUEBAS FINALES DEL CLÚSTER CON 3 NODOS.....	155
ANEXO 11	COMANDOS ÚTILES.....	158

## **LISTA DE FIGURAS**

FIGURA 1. COMPONENTES DE LA ESTRATEGIA DE IMPLEMENTACIÓN .....	32
FIGURA 2. REPRESENTACIÓN GRÁFICA DE UN ESCENARIO PROBLEMA .....	34
FIGURA 3. REPRESENTACIÓN GRÁFICA DE UN ESCENARIO SOLUCIÓN .....	36
FIGURA 4. CONFIGURACIÓN CON DISCO COMPARTIDO.....	41
FIGURA 5. CONFIGURACIÓN CON DISCOS EN CADA NODO.....	42
FIGURA 6. COMPONENTES DEL ADMINISTRADOR DE MEMBRESÍA Y MENSAJERÍA.....	43
FIGURA 7. DISEÑO DEL CLÚSTER CON 2 NODOS .....	47
FIGURA 8. COMPONENTES DE SOFTWARE .....	48
FIGURA 9. COMPONENTES IMPORTANTES DE UN RECURSO DRBD .....	55
FIGURA 10. COMPONENTES PRINCIPALES DE COROSYNC.....	60
FIGURA 11. COMPONENTES INTERNOS DE PACEMAKER.....	64
FIGURA 12. DISEÑO DEL CLÚSTER CON 3 NODOS.....	73
FIGURA 13. ACTIVIDADES DE CONFIGURACIÓN EN EL TERCER NODO .....	74
FIGURA 14. ACTIVIDADES PARA AGREGAR UN NODO ADICIONAL AL CLÚSTER.....	76
FIGURA 15. RESULTADOS DE SINCRONIZACIÓN-CLÚSTER CON 2 NODOS ..	79
FIGURA 16. RESULTADOS DE SINCRONIZACIÓN-CLÚSTER CON 3 NODOS .	83

## **TABLAS**

TABLA 1. ESCENARIOS PARA PRUEBAS DE REPLICACIÓN - CLÚSTER CON 2 NODOS .....	79
TABLA 2. PRUEBAS DE REPLICACIÓN DE ARCHIVOS - CLÚSTER CON 2 NODOS .....	80
TABLA 3. PRUEBAS DE MIGRACIÓN DE RECURSOS - CLÚSTER CON 2 NODOS .....	81
TABLA 4. ESCENARIOS PARA PRUEBAS DE REPLICACIÓN - CLÚSTER CON 3 NODOS .....	83
TABLA 5. PRUEBAS DE MIGRACIÓN DE RECURSOS - CLÚSTER CON 3 NODOS .....	84
TABLA 6. PARTICIONES DE DISCO .....	160

TABLA 7. CONFIGURACIÓN DE RED DEL NODO A .....	160
TABLA 8. CONFIGURACIÓN DE RED DEL NODO B .....	161
TABLA 9. CONFIGURACIÓN DE RED DEL NODO C .....	161

## **CAPITULO I**

### **1 . INTRODUCCION**

#### **1 . 1 . GENERALIDADES**

La idea de realizar este proyecto surgió al ver el impacto que representa la pérdida total de los servicios informáticos que por su naturaleza deberían funcionar las 24 horas del día, los 7 días de la semana, los 365 días del año en un ambiente comercial. Por otro lado, tenía el anhelo de explorar aún más el mundo del software libre, integrando ideas y programas en una solución práctica y accesible.

#### **1 . 2 . ANTECEDENTES**

Durante mi carrera universitaria tuve la oportunidad de participar en varios proyectos para empresas de nuestro país, lo que me permitió conocer de primera mano la infraestructura tecnológica con la cuentan estas empresas. Con este antecedente puedo decir que una gran cantidad de esas empresas, sin importar su tamaño, esperan que los sistemas informáticos se conviertan en una herramienta de utilidad para los empleados, de tal manera que se mejoren y

agilicen los procesos para que la empresa obtenga más beneficios.

Así mismo he visto que comúnmente las empresas pequeñas y medianas de nuestro país tienden a seleccionar alguna distribución de Linux para sus servidores porque no están obligadas a pagar licencias de uso como sucede con los productos propietarios, además porque en la mayoría de casos lo único que se necesitan es que el servidor Linux provea los servicios básicos para el acceso de navegación a internet, envío y recepción de correo electrónico, transmisión de archivos, y protección en el acceso de red. En otras palabras, un servidor Linux estándar para estas empresas cumple con las funciones de Servidor Proxy, Servidor de Correo (SMTP), Servidor de transferencia de Archivos (FTP) y Servidor Firewall. Obviamente, existen casos en donde el servidor Linux va más allá de las funciones básicas, como por ejemplo: servidor de aplicaciones, servidor de bases de datos, servidor web, etc.

Otra razón por la que las empresas de nuestro país tienden a seleccionar Linux como su Sistema Operativo es por su reputación de estabilidad y flexibilidad, es decir que un solo servidor es usado para cubrir una gran variedad de

necesidades en las empresas, desde las más básicas como las mencionadas anteriormente hasta necesidades específicas como el soporte para virtualización, telefonía IP, VPN, etc.

Sin lugar a dudas Linux posee características muy interesantes que han logrado atraer enormemente la atención de muchas empresas, no solo en nuestro país sino en todo el mundo. Sin embargo, son estas mismas características las que han creado un problema muy particular al que le ha dado el nombre de "el servidor todo en uno". Este problema consiste en que las empresas de nuestro país en su afán de reducir costos tienden a concentrar todos los servicios informáticos en un solo servidor, sin tomar en cuenta que cuando éste falle o necesite ser detenido por tareas de mantenimiento [preventivo-correctivo] la empresa prácticamente tendrá que paralizar las actividades que dependan directamente de los servicios que ofrece el servidor porque los usuarios no tendrán acceso a las herramientas de trabajo y eso ocasionará pérdidas económicas para la empresa.

Con todos estos antecedentes podemos ver que si una empresa basa o confía sus operaciones en los servicios que provee un solo servidor automáticamente tiene un serio problema de redundancia y a la vez una importante necesidad de alta

disponibilidad en los servicios y aplicaciones de internet e intranet, razón por lo cual propongo como parte de la solución a este tipo de problema --**el servidor todo en uno**-- una estrategia para la implementación de un Clúster de Alta Disponibilidad utilizando software libre GNU/Linux para ayudar a mitigar el problema de redundancia y cubrir la necesidad de alta disponibilidad de tal manera que los usuarios, empleados, ejecutivos, gerentes, accionistas, etc. de la empresa o empresas que decidan optar por esta solución tengan acceso continuo a los servicios de intranet y extranet cuando se presentan problemas con el servidor principal.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO FINAL**

Este proyecto tiene como objetivo final proveer una estrategia para la implementación de un Clúster de Alta Disponibilidad usando software de distribución libre bajo la licencia GNU/Linux.

### **1.3.2. OBJETIVOS ESPECIFICOS**

- Plantear una estrategia para la implementación de un Clúster de Alta disponibilidad.
- Implementar un prototipo de un Clúster de Alta Disponibilidad sobre la plataforma Linux CentOS.
- Configurar e Integrar dos nodos al Clúster de Alta Disponibilidad con 2 Servicios (FTP y DNS).
- Configurar e Integrar un tercer nodo al Clúster de Alta Disponibilidad con 1 Servicio adicional (HTTP).

### **1.4. JUSTIFICACION**

En la actualidad los servicios de computación relacionados con el Internet como el correo electrónico, la navegación Web y transferencia de archivos son una parte importante en el día a día de muchas empresas, por esta razón es necesario implementar mecanismos de redundancia que permitan que estos servicios estén disponibles la mayor cantidad de tiempo posible para los usuarios.

Por esta razón se justifica la implementación de este proyecto en vista de la necesidad de mejorar la redundancia y la alta disponibilidad de aquellas empresas que dependen de los servicios proporcionados por un solo servidor.

Por otro lado es importante que las empresas de nuestro país utilicen tecnología de punta dentro de su infraestructura tecnológica para poder competir con otras empresas del país y del mundo.

### **1.5. ALCANCE**

El alcance de este proyecto se limita al planteamiento de una estrategia para la implementación de un Clúster de Alta Disponibilidad usando herramientas y aplicaciones de distribución libre para el Sistema Operativo Linux. La propuesta contenida en esta tesis proveerá los lineamientos básicos de configuración de un clúster de N nodos que luego pueden ser profundizados, perfeccionados y adaptados a las necesidades particulares de las empresas o instituciones que deseen implementarlo. Sin embargo, el número total de nodos del prototipo del Clúster desarrollado para demostrar la funcionalidad de la estrategia propuesta será de 3 nodos.

### **1.6. ASPECTOS METODOLOGICOS**

Internet es la fuente más grande e importante para la distribución del software libre, es por eso que este proyecto se basa en gran parte en la información publicada

en libros electrónicos, foros de discusión, blogs, documentos, revistas digitales y páginas web encontradas en Internet.

Para poder implementar la estrategia planteada en este proyecto se llevarán a cabo las siguientes actividades:

- Identificación y Localización del Software
- Evaluación y Selección del Software
- Obtención del Software y su Documentación.
- Instalación y Configuración del Software
- Pruebas

#### **1.6.1. IDENTIFICACIÓN Y LOCALIZACIÓN DEL SOFTWARE**

La Identificación del software y su posterior localización son dos de los puntos más importantes para este tipo de proyectos ya que de manera general existe una cantidad importante de herramientas y aplicaciones de distribución libre prácticamente desconocidas.

Estas actividades se realizan principalmente con la ayuda de referencias obtenidas en motores de búsqueda, páginas Web, foros de discusión, revistas, etc.

### **1.6.2. EVALUACIÓN Y SELECCIÓN DEL SOFTWARE**

Por evaluación se entiende a la obtención de información acerca del software, esta actividad permite tener referencias de los programas posteriormente seleccionados. En estas actividades se consideran varios aspectos tales como: usabilidad, utilidad, características, soporte, costo de implementación, referencias comparativas realizadas por otras personas, etc.

### **1.6.3. OBTENCIÓN DEL SOFTWARE Y SU DOCUMENTACIÓN**

La forma más sencilla de obtener software libre y su respectiva documentación (si existe) es a través del Internet aunque también es posible hacerlo a través de medios físicos como una memoria flash, un CD o un DVD ya sean adquiridos o copiados (¡el software libre lo permite!).

Cabe mencionar que si bien la mayoría de estos programas incluyen breves manuales de instalación, en la mayoría de casos es necesaria la ayuda de documentación adicional.

#### **1.6.4. INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE**

Sin lugar a dudas estos son los dos puntos más importantes dentro de la elaboración de este tipo de proyectos por ello es recomendable leer y entender la documentación incluida con el software para que la instalación y la configuración seleccionada sea la más adecuada.

También es importante cumplir los requerimientos individuales de cada programa para evitar problemas de incompatibilidad.

#### **1.7. REQUERIMIENTOS BÁSICOS**

- 3 Computadores con tarjetas de red Ethernet y disco duro.  
(Pueden ser máquinas virtuales).
- 1 Red Ethernet.
- Acceso a Internet.

#### **1.8. RENUNCIA DE RESPONSABILIDAD**

TODA LA INFORMACIÓN CONTENIDA EN ESTA TESIS ES PROVISTA POR PARTE DEL AUTOR **COMO ESTÁ**, SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA. EN NINGÚN CASO EL AUTOR DE ESTA TESIS SERÁ RESPONSABLE POR DAÑOS DIRECTOS, INDIRECTOS O CUALESQUIERA QUE ESTOS FUEREN Y QUE RESULTEN, SIN RESTRICCIONES, EN

PÉRDIDA DE USO, DATOS, VENTAS, GANANCIAS, INTERRUPCIÓN DEL NEGOCIO, PÉRDIDA DE PROGRAMAS U OTROS DATOS SOBRE CUALQUIER SISTEMA DE MANEJO DE LA INFORMACIÓN O BAJO OTRAS CIRCUNSTANCIAS, SEAN EN UN CONTRATO, NEGLIGENCIA U OTRA ACCIÓN INCORRECTA, YA SEA A CAUSA DEL ACCESO A, AL USO O EN CONEXIÓN CON EL USO DE, INHABILIDAD DE USAR, EXACTITUD O FACTIBILIDAD DE USO DE ESTA INFORMACIÓN O DE LOS PROGRAMAS Y CUALQUIER REFERENCIA BIBLIOGRÁFICA, INCLUYENDO, LA PÉRDIDA DE EMPLEO Ó TRABAJO, INCLUSO SI EL AUTOR ES NOTIFICADO EXPRESAMENTE DE LA POSIBILIDAD DE TALES DAÑOS.

## CAPITULO II

### 2. DEFINICIONES Y CONCEPTOS IMPORTANTES

#### 2.1. CLÚSTER

Aunque el concepto de clúster en términos informáticos parece sencillo realmente no lo es. Por ejemplo, en el Libro "In Search of Clusters" de Gregory F. Pfister el autor tiene dificultades para establecer una definición única y exacta para el término clúster, pese a ello el autor en la página 72 llega a determinar que un clúster es "un tipo de sistema distribuido o paralelo conformado por una colección de computadoras interconectadas usado como un único recurso de computación unificado" (Pfister, 1998).

Así mismo, reconocidos expertos en este tema como Avi Silberschatz y Peter Galvin en su libro "Operating System Concepts" definen a un clúster como "un conjunto de máquinas y un servidor dedicado, para realizar los relativamente infrecuentes accesos a los recursos de otros procesos" (Silberschatz, Galvin, & Gagne, 2005). Para terminar de mostrar la disparidad de opiniones que existen en la definición de un clúster veamos un resumen de la definición dada por los Doctores Kai Hwang y Khiwei Xu en su libro

"Scalable Parallel Computing", "un clúster es la variación de bajo precio de un multiprocesador masivamente paralelo (miles de procesadores, memoria distribuida, red de baja latencia), con las siguientes diferencias: cada nodo es una máquina quizás sin algo del hardware (monitor, teclado, mouse, etc.) conectados por una red (preferible de bajo precio como Ethernet) cuya interfaz de red no está muy acoplada al bus I/O y donde todos los nodos tienen disco local con su respectivo sistema operativo con una capa de software capaz de soportar todas las características del clúster" (Hwang & Xu, 1998).

Como podemos ver no existe una definición exacta sobre lo que es un clúster, sin embargo por razones de simplicidad y para evitar una discusión que puede requerir de varias páginas, en esta tesis la definición de clúster se entenderá de manera general y en términos simples como un grupo de máquinas interconectadas trabajando por un servicio conjunto de manera coordinada. De ningún modo esta definición pretende ser estricta pues ¿qué sucedería si conectamos 2 consolas de video juegos con el objetivo de jugar en red? ¿acaso no es eso un clúster? O que tal si en lugar de usar las consolas para jugar las usamos para realicen procesamiento paralelo ¿no es eso también un clúster?

Ahora que he resaltado que no hay un acuerdo unánime acerca de la definición de un clúster no me queda más que aclarar que cuando se utilice el término clúster en este documento, las similitudes con un sistema distribuido o incluso con un sistema paralelo no son intencionales y por lo tanto no serán discutidas.

### **2.1.1. TIPOS DE CLUSTERS**

Los clústeres cuyos nodos tienen arquitecturas similares y un mismo sistema operativo se los conoce como **Homogéneos** caso contrario se los conoce como **Heterogéneos**.

Así mismo, existen 3 áreas principales en las que los clústeres tienen su campo de acción y es precisamente de allí de donde toman su nombre.

- Clústeres de Balanceo de Carga.
- Clústeres de Alta Disponibilidad.
- Clústeres de Alto Rendimiento.

#### **2.1.1.1. Clústeres de Balanceo de Carga.**

El objetivo de un clúster de Balanceo de Carga es el de distribuir el trabajo entre todos los nodos del

clúster, esto se consigue asignando el trabajo al nodo que posee más recursos disponibles. Este tipo de clústeres se usan comúnmente en servicios que por su naturaleza tienen una enorme cantidad de carga, como por ejemplo: los servidores Web.

#### **2.1.1.2. Clústeres de Alto Rendimiento.**

El objetivo de un clúster de alto rendimiento es compartir el recurso más valioso de un computador, es decir, la capacidad de procesamiento. Este tipo de clústeres son muy populares, y quizás también los más antiguos, ya que en sus inicios fueron desarrollados para centros de computo enfocados en investigaciones científicas donde era necesario tener un gran poder de procesamiento, capaz de soportar una inmensa cantidad de cálculos para resolver un problema específico.

#### **2.1.1.3. Clústeres de Alta Disponibilidad.**

El objetivo principal de un clúster de alta disponibilidad, también conocido como clúster HA por las palabras en inglés High Availability, es el de mantener disponible uno o más servicios la mayor

cantidad de tiempo posible. Este tipo de clústeres se caracterizan por que sus nodos están monitoreándose de manera continua, es así que, cuando se detecta un falló de hardware o software en uno de los nodos, otro nodo se encarga de proveer los servicios del nodo con problemas, minimizando así la percepción de la pérdida de servicio por parte de los usuarios.

En vista que el objetivo de este proyecto es mostrar una estrategia para la implementación de un clúster de alta disponibilidad es necesario profundizar algunos conceptos y fundamentos relacionados a este tipo de clústeres.

### **2.1.2. COMPONENTES PRINCIPALES DE UN CLÚSTER DE ALTA DISPONIBILIDAD**

Los componentes principales de un clúster HA son:

- Nodos
- Sistema Operativo
- Conexiones de Red
- Middleware
- Recursos o Servicios

### **2.1.2.1. NODOS**

Un nodo no es otra que un computador dentro del clúster. Un nodo puede ser un computador convencional, un servidor o un súper computador. Aunque no es estrictamente obligatorio que todos los nodos del clúster sean exactamente iguales es recomendable que estos compartan características similares de procesamiento y almacenamiento para evitar un comportamiento ineficiente del clúster.

### **2.1.2.2. SISTEMA OPERATIVO**

Una definición más precisa de lo que es un sistema operativo se encuentra más adelante, sin embargo se puede decir de manera general que un sistema operativo es aquel que se encarga de alojar los recursos e interactuar con el hardware en un computador. No todos los sistemas operativos disponibles tienen las características necesarias para conformar un clúster, un claro ejemplo de esta realidad es MS-DOS. Por lo tanto, un sistema operativo para clústeres debe ser robusto, estable y confiable, además debe ser multiusuario y multiproceso.

#### **2.1.2.3. CONEXIONES DE RED**

Computacionalmente hablando, una conexión de red es el medio por el cual, al menos dos computadores, envían y reciben información. Un requisito indispensable para la implementación de un clúster es que todos los nodos sean capaces de comunicarse entre sí a través de una conexión de red. Independientemente de la tecnología seleccionada se recomienda que la velocidad de comunicación entre los nodos sea lo más homogénea posible y mientras más rápida, mejor.

#### **2.1.2.4. MIDDLEWARE**

De acuerdo con el profesor Sacha Krakowiak de la Universidad Joseph Fourier de Francia, un middleware es "...la capa de software que se encuentra entre el sistema operativo y las aplicaciones en cada sitio del sistema." (**Krakowiak, 2003**) En otras palabras, es un programa intermediario entre dos programas independientes. El middleware es el encargado de monitorear, administrar, distribuir y controlar el comportamiento general del clúster.

Algunos de los middlewares de alto nivel más populares para clústeres son:

- Beowulf.
- OSCAR (Open Source Cluster Application Resources).
- OpenMosix.
- Pacemaker.
- Red Hat HPC Solution cluster.

#### **2.1.2.5. RECURSOS O SERVICIOS**

Desde el punto de vista de un administrador de sistemas, un servicio o aplicación son piezas de código que se ejecutan en el servidor para cumplir una tarea específica. Sin embargo desde el punto de vista del Clúster, un servicio o aplicación son simples recursos que pueden ser iniciados o detenidos. Por ejemplo, si hablamos de un servicio Web ó un servidor HTTP, el recurso para el clúster es el servidor Web. De la misma manera, para un servicio de correo o servidor SMTP, el recurso es el servidor de correo. Esto también quiere decir que todo recurso tiene un servicio asociado.

### 2.1.3. MODELOS DE IMPLEMENTACIÓN

Los clústeres de Alta Disponibilidad requieren de al menos dos nodos para proveer redundancia, sin embargo muchas implementaciones pueden tener N nodos. Tomando en cuenta esta característica los modelos de implementación de un clúster pueden variar de la siguiente manera:

- **Activo/Activo:** Este modelo también se conoce como simétrico y consiste en reenviar el tráfico dirigido al nodo con problemas a otro nodo del clúster. Para implementar este modelo es necesario que todos los nodos tengan una configuración homogénea.
- **Activo/Pasivo:** Este modelo también se conoce como asimétrico. Cada nodo tiene un nodo de respaldo, esta configuración requiere más hardware.
- **1-a-N:** Un solo nodo del clúster provee todos los servicios y cuando este falla, esos servicios son transferidos temporalmente al resto de nodos que normalmente son nodos más pequeños o con menos recursos.
- **N-a-1:** Solo un nodo del clúster toma temporalmente el lugar del nodo con problemas.

- **N-a-N:** Varios o todos los nodos del clúster proveen servicios y cuando uno de ellos falla, sus servicios se transfieren temporalmente a cualquier otro nodo.

#### **2.1.4. ARQUITECTURA**

Los clústeres de Alta Disponibilidad pueden o no tener una unidad de almacenamiento compartido. Comúnmente, las configuraciones de conexión de los subsistemas almacenamiento compartido se realizan a través de canales de fibra de alta velocidad, cables, intercambiadores o iSCSI.

Una ventaja particular de esta arquitectura es que si un servidor falla, otro servidor designado automáticamente tiene acceso continuo a los archivos y directorios del disco compartido. Por otro lado, esta arquitectura podría crear un único punto de fallo si el medio de almacenamiento falla.

En el caso de compartir una unidad de almacenamiento es recomendable, usar un sistema de archivos para clústeres.

### 2.1.5. SISTEMA DE ARCHIVOS PARA CLÚSTERES

Un sistema de archivos para clústeres es aquel que permite el acceso compartido y simultáneo de datos a múltiples servidores pero al mismo tiempo provee la suficiente capa de protección para prevenir corrupción e inconsistencia de datos. A menudo, un sistema de archivos para clústeres es un elemento crítico en la operación y funcionamiento de un clúster, ya que a diferencia de la mayoría de los sistemas de archivos convencionales, este sistema de archivos está diseñado para que aquellas aplicaciones que se ejecutan en múltiples nodos del clúster tengan acceso óptimo a archivos concurrentes, es decir, que varios servidores pueden acceder a un archivo al mismo tiempo.

Entre las ventajas más importantes que provee un sistema de archivos para clústeres están:

- Permitir el acceso a un mismo dispositivo de almacenamiento en modo lectura/escritura.
- Realizar checksums<sup>1</sup> de los metadatos para prevenir la corrupción de datos.

---

<sup>1</sup> Véase el Glosario

- Poseer una o más capas para la comunicación intrínseca del clúster.
- Permitir la indexación de directorios.
- Soportar E/S en todos los nodos para mayor rendimiento y flexibilidad.

Algunos de los sistemas de archivos para clústeres más comunes son:

- IBM General Parallel File System (GPFS) .
- HP Cluster File System (CFS) .
- Oracle Cluster File System (OCFS y OCFS2) .
- Global File System (GFS) .
- Clustered XFS (CXFS) .

## **2.1.6. OTRAS DEFINICIONES**

### **2.1.6.1. ALTA DISPONIBILIDAD**

La disponibilidad se basa en un modelo matemático que provee el grado en que un sistema de software permanece en una condición operable.

$$A = \frac{MTTF}{MTTF + MTTR}$$

MTTF = Es el tiempo promedio entre fallos.

MTTR = Es el tiempo promedio que tomar reparar el sistema después de un fallo.

Existen 2 métodos para alcanzar una alta disponibilidad.

- Incrementar el valor del MTTF.
- Reducir el valor del MTTR.

#### **2.1.6.2. FAILOVER**

Se conoce como Failover al proceso de migración de un servicio o aplicación desde un nodo del clúster a otro.

Este proceso se puede realizar manualmente, sin embargo, en condiciones ideales se realiza automáticamente.

##### **2.1.6.2.1. POLÍTICAS DE FAILOVER**

En el caso de un clúster de dos nodos, la política de failover es simple y casi obvia, ya que la única

opción disponible es transferir todos los servicios del nodo con problemas al nodo saludable.

- **N+1:** Varios servicios son transferidas del nodo o nodos con problemas a un nodo saludable.
- **N+M:** Varios servicios son transferidos del nodo o nodos con problemas a un nodo saludable que no esté proveyendo ningún servicio.
- **Failover-Ring:** Todos los nodos del clúster son etiquetados secuencialmente, esto permite transferir todos los servicios de un nodo con problemas al siguiente nodo en secuencia.

#### **2.1.6.3. REDUNDANCIA**

Los clústeres de alta disponibilidad implementan el concepto de redundancia (**Sloan, 2004**). La Redundancia consiste en introducir un dispositivo de respaldo en un sistema existente, con la finalidad de asegurar el servicio ininterrumpido del sistema a pesar de un fallo de hardware o software.

#### **2.1.6.4. APPLICACIONES DE MISIÓN CRÍTICA**

Las aplicaciones de misión crítica son aquellas que por su naturaleza deben estar operativas las 24 horas del día, los 365 días del año. En este tipo de aplicaciones una falla debe pasar casi desapercibida para los usuarios.

#### **2.1.6.5. MEMBRECÍA**

La membrecía es el proceso mediante el cual un nodo obtiene acceso al clúster.

#### **2.1.6.6. MENSAJERÍA**

La mensajería es el proceso mediante el cual los nodos se comunican dentro del clúster.

#### **2.1.6.7. CERCADO**

El cercado es el mecanismo que se usa para desconectar un nodo con problemas del clúster cuando se usa un disco compartido.

## **2.1.7. BENEFICIOS DE UN CLÚSTER DE ALTA DISPONIBILIDAD**

Uno de los beneficios principales de un clúster de alta disponibilidad es la reducción de las interrupciones de los servicios por problemas imprevistos de hardware o software, ya que los recursos pueden ser configurados de tal manera que sean migrados automáticamente a otro servidor ante un evento ocasionado por problemas en el servidor principal. Así mismo es posible cambiar o mover los recursos dinámicamente entre los nodos del clúster cuando existen tareas planificadas de mantenimiento o actualizaciones de hardware y software.

Otros beneficios de un clúster de alta disponibilidad son:

- Mayor disponibilidad
- Bajo costo de operación
- Escalabilidad
- Recuperación de desastres
- Protección y Redundancia de datos

## 2.2. SISTEMA OPERATIVO

En el libro “Informática Uno” Martha Angélica Orozco define de manera sencilla a un Sistema Operativo como:

“...un programa que actúa como intermediario entre el usuario y el hardware de una computadora y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas. El objetivo principal de un Sistema Operativo es lograr que el sistema de computación se use de manera cómoda, y el objetivo secundario es que hardware de la computadora se emplee de manera eficiente.” (**Orozco, 2006**)

### 2.2.1. SISTEMA OPERATIVO GNU/LINUX

GNU/Linux es un Sistema Operativo Abierto desarrollado y soportado por muchas empresas y personas alrededor del mundo. En términos generales, Linux es muy similar a otros sistemas operativos como Microsoft Windows o Mac OS, sin embargo como mencionan Neil Matthew y Richard Stones en su libro “Beginning Linux Programming”, Linux es únicamente el kernel(4) y se complementa con muchas librerías, aplicaciones y herramientas de desarrollo distribuidas

libremente bajo la Licencia Pública General GNU. La Combinación de GNU y Linux forman los Sistemas Operativos GNU/Linux, también conocidos como distribuciones GNU/Linux o [también mal llamados] simplemente "Linux" (**Matthew & Stones, 2011**).

#### **2.2.1.1. PROYECTO GNU**

El Proyecto GNU fue lanzado en 1984 con la finalidad de desarrollar un Sistema Operativo Unix-like cuya traducción al español significa como-Unix.

#### **2.2.1.2. LICENCIA PÚBLICA GENERAL GNU**

La Licencia Pública General GNU fue creada en 1989 por la Fundación de Software Libre (Free Software Foundation) con el objetivo principal del proteger la libre distribución, modificación y uso del software. El término GNU es el acrónimo recursivo de "GNU no es Unix!" (GNU's Not Unix!). La Licencia Pública General GNU comúnmente se entiende como un acuerdo o contrato de licencia pues es un documento que cede determinados derechos al usuario.

### 2.2.1.3. DISTRIBUCIONES GNU/LINUX

Algunas de las distribuciones de Sistemas GNU/Linux más populares son: (**Tony Bautts, 2005**)

- Red Hat Enterprise Linux.
- Debian GNU/Linux.
- Suse Linux Enterprise Server.
- Ubuntu.
- CentOS (Community Enterprise Operating System) .
- Fedora.

### 2.3. SERVIDOR WEB

Un servidor Web es un sistema que se encarga de atender y responder las peticiones que usan el protocolo de transferencia de hipertexto para visualizar las páginas Web. Un servidor Web básico cumple con el siguiente esquema de funcionamiento.

- Espera por peticiones Web en el puerto TCP indicado (el estándar por defecto para HTTP es el 80) .

- Recibe una petición Web.
- Busca el recurso solicitado.
- Envía el recurso utilizando la misma conexión por la que recibió petición.

#### **2.4. SERVIDOR DE NOMBRES DE DOMINIO**

Un servidor de nombres de dominio (DNS) es un sistema que permite asignar nombres a equipos o servicios de red. Por ejemplo las redes TCP/IP, como Internet, usan un DNS para buscar equipos y servicios mediante nombres descriptivos que son mucho más fáciles de recordar que las direcciones numéricas que los equipos usan para comunicarse a través de Internet. Cuando un usuario escribe un nombre DNS descriptivo en una aplicación como un navegador web, el servidor DNS convierte el nombre a su dirección numérica correspondiente.

#### **2.5. SERVIDOR DE TRANSFERENCIA DE ARCHIVOS**

Un servidor de transferencia de archivos es un sistema que permite transferir archivos entre computadores en una red

TCP. Para lograr este objetivo se ha definido el protocolo de transferencia de archivos que es el encargado de controlar el proceso de transferencia desde el origen hacia el destino.

## CAPITULO III

### 3. ESTRATEGIA DE IMPLEMENTACION

La estrategia de implementación propuesta consiste en instalar, configurar e integrar todos los componentes que se muestran en la figura 1.

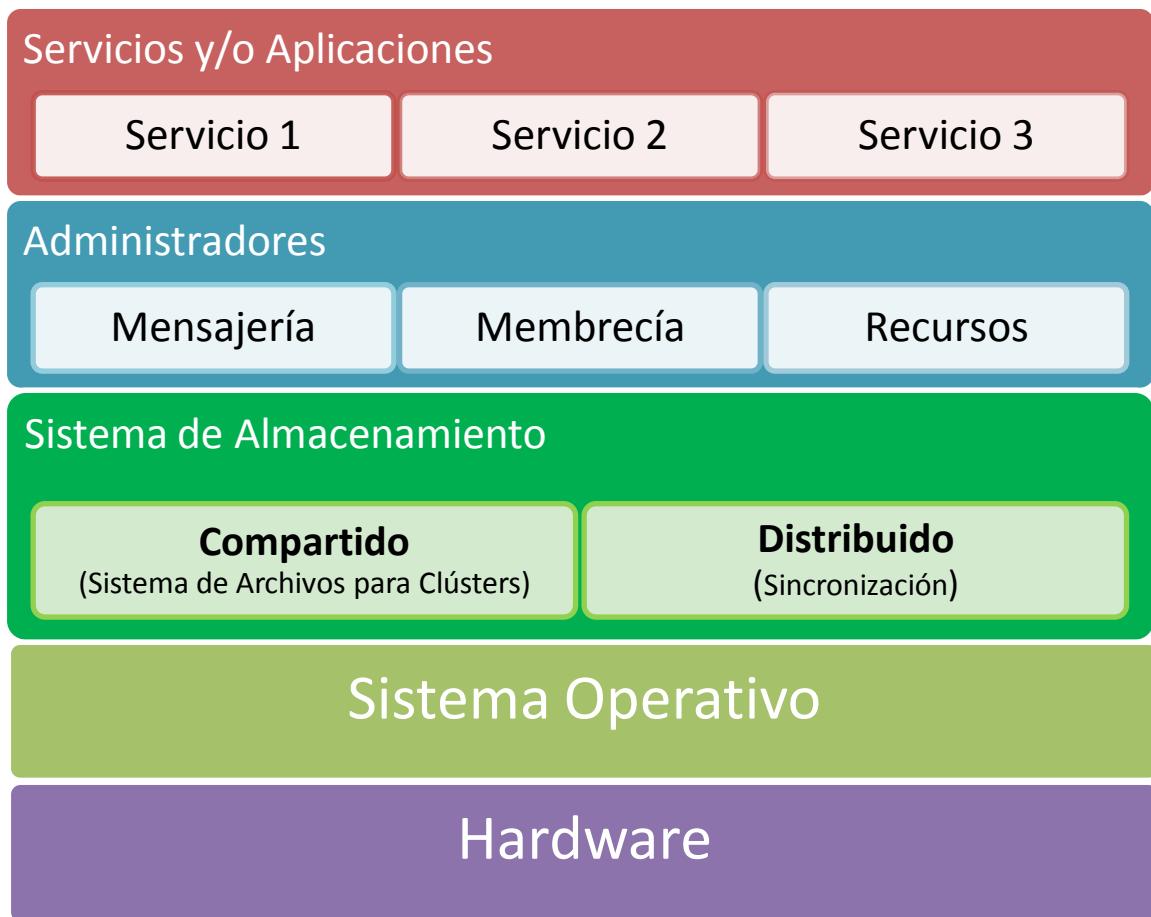


Figura 1. – Componentes de la Estrategia de Implementación.

Para completar la estrategia es necesario realizar las siguientes actividades:

- Diseñar la arquitectura física del clúster.
- Instalar y configurar el sistema operativo GNU/Linux.
- Instalar y configurar el sistema de archivos para clústeres ó el administrador de sincronización de datos.
- Instalar el administrador de membresía y mensajería.
- Instalar y configurar el administrador de recursos.
- Instalar y configurar los servicios y/o aplicaciones.

### **3.1. DISEÑO DEL CLÚSTER**

La primera actividad consiste en diseñar la arquitectura del clúster. El diseño del clúster permite visualizar de manera gráfica la arquitectura física de la solución que se pretende implementar, así mismo, permite definir el propósito de la solución. Antes de diseñar la arquitectura del clúster es necesario revisar y entender claramente el problema que se va resolver. Por ejemplo, imaginemos que la empresa SGP tiene la arquitectura de red de la Figura 2.

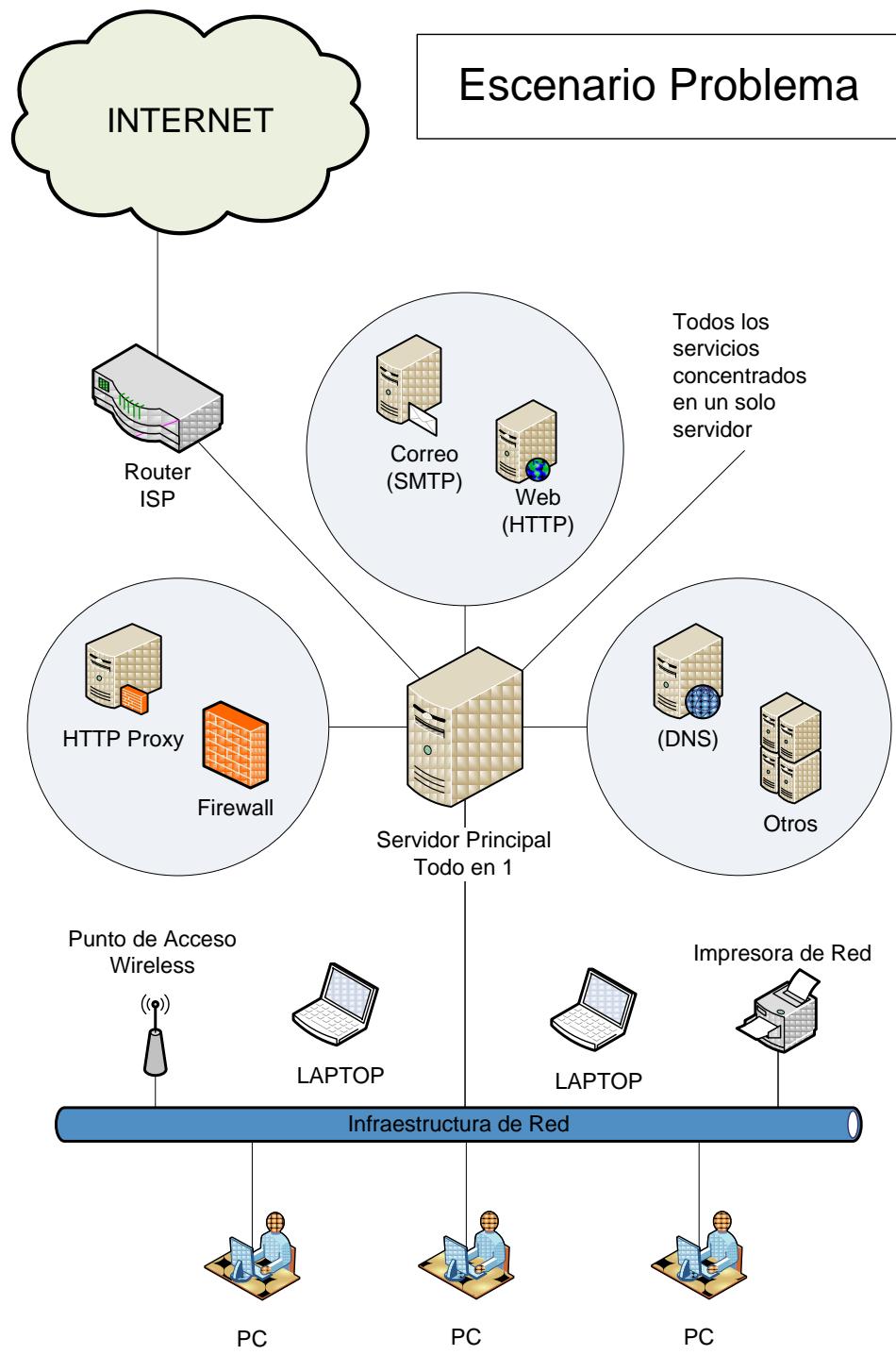


Figura 2. Representación gráfica de un escenario problema.

Un solo servidor principal con todos los servicios.

De manera general esta representación gráfica puede resumirse de la siguiente manera: Existe un servidor principal que aloja uno o más servicios informáticos, el cual no tiene ningún tipo de redundancia, por lo tanto cada vez que éste presenta un mal funcionamiento o no está trabajando afecta significativamente a las actividades que dependen de ese servidor.

Una vez entendido el problema también es necesario entender la solución que se va implementar. De manera general la solución del problema puede resumirse de la siguiente manera: Implementar un Clúster de Alta Disponibilidad de 2 nodos con el objetivo de proveer redundancia al servidor principal, de tal manera que un servidor secundario sea capaz de reemplazar al servidor principal cuando este presenta problemas.

Con esta información es posible realizar una representación gráfica del diseño del clúster que será implementado, como se muestra en la Figura 3.

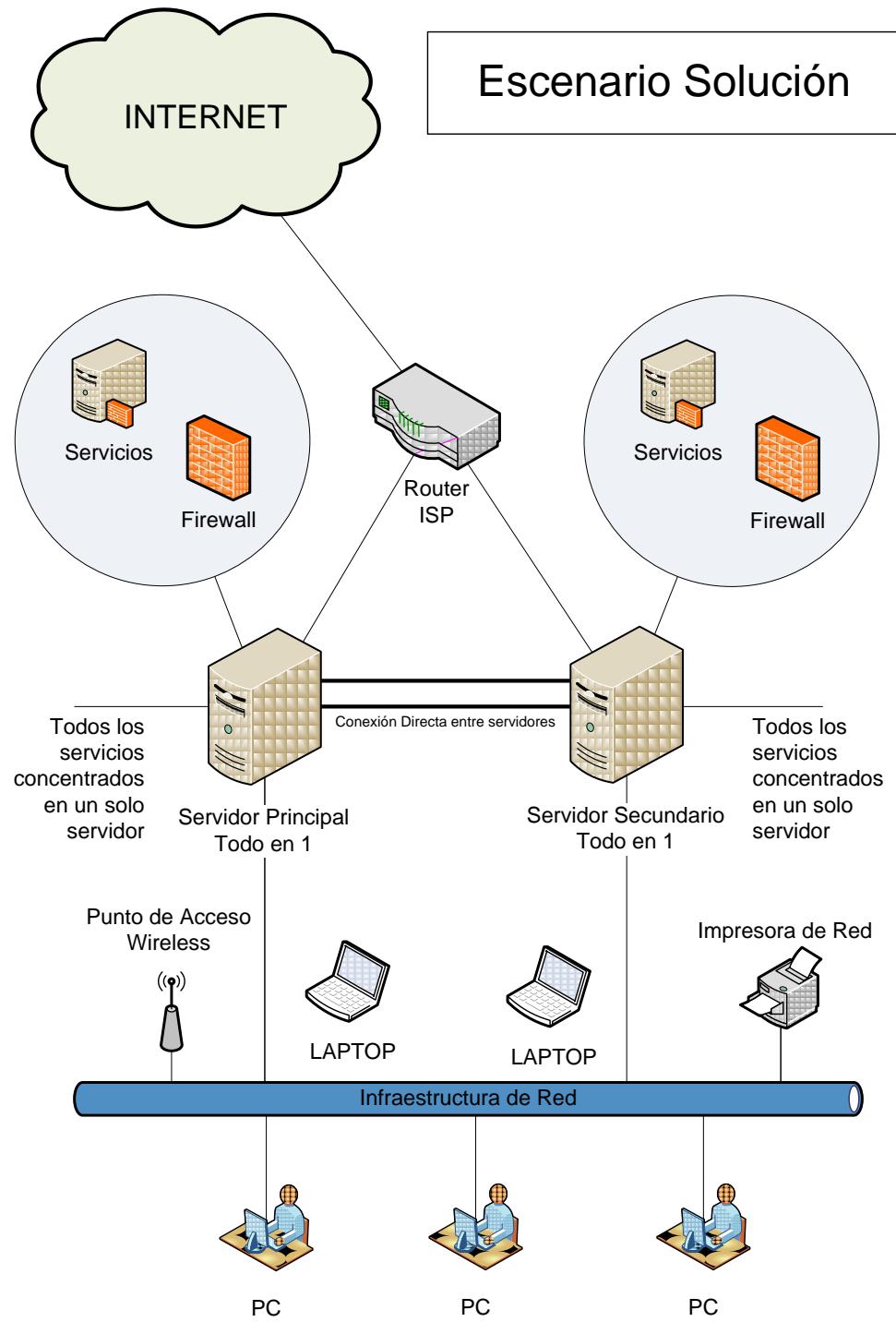


Figura 3. Representación gráfica de un escenario solución.

Dos servidores, uno principal y otro secundario con todos los servicios.

### **3.2. INSTALACIÓN DEL SISTEMA OPERATIVO**

La segunda actividad consiste en instalar una distribución GNU/Linux. Dentro del clúster, el sistema operativo es el encargado de proveer la plataforma sobre la que se ejecutan las aplicaciones, los servicios, los administradores de mensajería, membresía y sincronización; por esta razón es recomendable instalar una distribución Linux<sup>2</sup> que sea estable, segura, robusta y ampliamente configurable, es decir que esté orientada a profesionales o ambientes empresariales. Algunas distribuciones que podrían cumplir estas características son: Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS, Oracle Linux ó Ubuntu Server.

#### **3.2.1. ACTUALIZACIONES DEL SISTEMA OPERATIVO**

Ignorar las notificaciones de actualización y mantener los sistemas Linux desactualizados es un error común entre los administradores de sistemas, ya que argumentan que las distribuciones Linux no las necesitan porque supuestamente no son vulnerables a virus y lo único que hacen es ralentizar el sistema. Lo que estos administradores no

---

<sup>2</sup> Véase Glosario - Distribución Linux

toman en cuenta es que las actualizaciones del sistema no solo corrigen problemas de seguridad o de programación que pueden ser aprovechados por software malintencionado o virus, sino que también dan soporte a nuevas tecnologías cuyo objetivo principal es mantener el sistema más estable.

Una vez mencionados algunos aspectos importantes de las actualizaciones en un sistema Linux también es importante mencionar algunas consideraciones y recomendaciones, a tomar en cuenta, con respecto a las actualizaciones: las siguientes recomendaciones:

- Instalar las actualizaciones como parte del proceso de instalación del sistema operativo.
- Planificar la actualización del sistema de manera periódica.
- Revisar y probar las actualizaciones en ambientes de prueba antes de ser aplicadas en ambientes de producción.
- Evitar las actualizaciones desatendidas pues esto puede ocasionar desde cortes de servicio inesperados hasta el reinicio total del sistema.

- Desactivar los procesos automáticos de actualización pero nunca omitir las notificaciones de actualizaciones disponibles.

### **3.2.2. RED LAN DE COMUNICACIÓN TCP/IP**

Un clúster simplemente no puede existir si sus nodos no pueden comunicarse entre sí. En Linux el soporte de los protocolos TCP/IP está prácticamente asegurado dado que la mayoría de las redes TCP/IP usan Ethernet y Linux tiene soporte para una larga lista de adaptadores de red de distintos fabricantes.

### **3.3. INSTALACIÓN DEL SISTEMA DE ARCHIVOS PARA CLÚSTERES Ó DEL SISTEMA DE ALMACENAMIENTO DISTRIBUIDO**

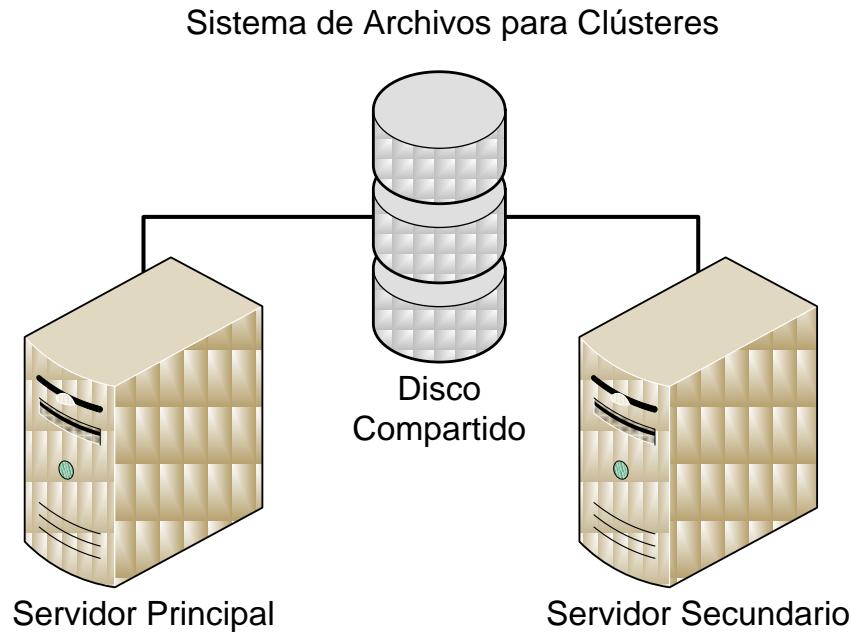
La tercera actividad de la estrategia consiste en instalar un sistema de archivos para clústeres o un sistema de almacenamiento distribuido. La instalación de un sistema de archivos para clústeres o de un sistema de almacenamiento distribuido depende directamente del diseño de clúster. Por ejemplo, el sistema de archivos para clústeres es un componente indispensable en la estrategia de implementación de un clúster si se usa almacenamiento compartido; no

obstante, es un componente opcional si cada nodo en el clúster posee su propio medio de almacenamiento.

### **3.3.1. SISTEMA DE ARCHIVOS PARA CLÚSTERES.**

De manera general, la configuración de un sistema de almacenamiento que permita a múltiples computadores acceder a un mismo dispositivo de bloques lógico es relativamente fácil, sin embargo, si el sistema no es configurado correctamente existe un riesgo muy grande de destruir la información almacenada, especialmente en el caso que dos o más computadores intenten usar el dispositivo al mismo tiempo; es por esta razón que un sistema de archivos para clústeres en medios de almacenamiento compartidos es prácticamente obligatorio.

En la figura 4 se muestra un ejemplo de una configuración con disco compartido.



**Figura 4. Configuración con Disco Compartido.**

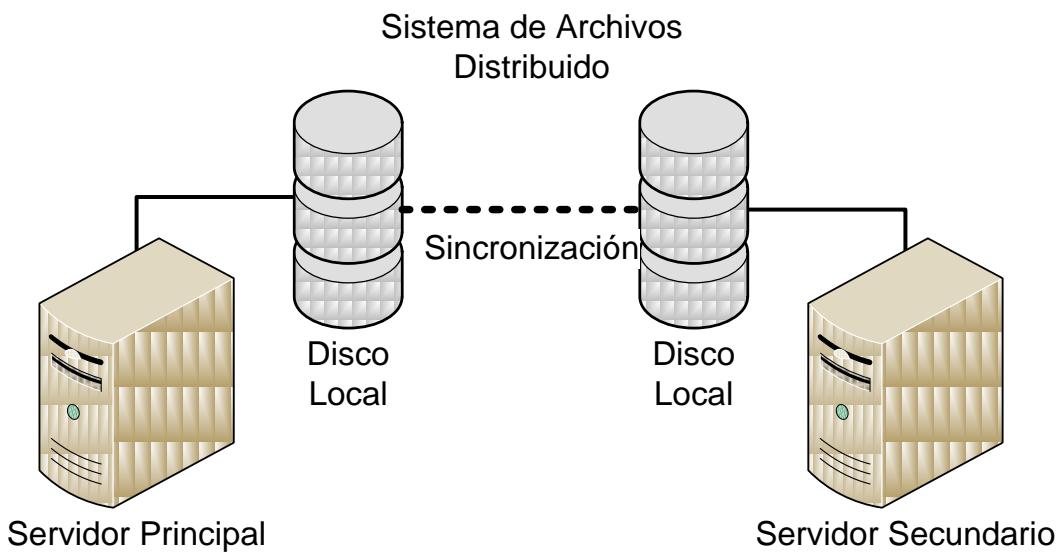
Algunos de los sistemas de archivos para clústeres más utilizados son: GFPS de IBM, CFS de HP, OCFS2 de Oracle o GFS de Red Hat.

### **3.3.2. SISTEMA DE ALMACENAMIENTO DISTRIBUIDO.**

También es posible configurar el clúster de tal manera que cada nodo posea su propio dispositivo de almacenamiento. Este tipo de configuración provee redundancia de datos al mantener múltiples copias en diferentes nodos; pero a su vez, requiere obligatoriamente de un sistema de

almacenamiento distribuido que actúe como administrador de sincronización para que notifique a los otros nodos cuando hay cambios en el servidor principal.

En la figura 5 se muestra un ejemplo de una configuración que usa un sistema de archivos distribuido.



**Figura 5. Configuración con discos en cada nodo.**

Algunos de los sistemas de almacenamiento distribuido más utilizados son: NBD, DRBD, GlusterFS, ExaFS ó MogileFS.

### **3.4. INSTALACIÓN DEL ADMINISTRADOR DE MEMBRECÍA Y MENSAJERÍA.**

La cuarta actividad consiste en instalar un administrador de membresía y un administrador de mensajería. El administrador

de membresía es el encargado de coordinar toda la información relacionada con los nodos del clúster, de tal manera que sea posible determinar los nodos activos y los que están fuera de servicio. Para poder alcanzar esta coordinación el administrador de membresía monitorea los mensajes enviados por los nodos y actualiza constantemente el estado de los servicios y las aplicaciones de cada nodo. Cuando la membresía del clúster cambia, el administrador de membresía notifica a todos los otros componentes de la infraestructura para que tomar la acción correspondiente.

En la figura 6 se muestra los componentes principales del administrador de Membresía y mensajería.



**Figura 6. Componentes del Administrador de Membresía y Mensajería.**

El administrador de mensajería es el encargado de proveer la capa de comunicación entre los nodos del clúster. Entre otras cosas este administrador provee el soporte de encriptación<sup>3</sup> y autenticación<sup>4</sup> para una comunicación segura entre los miembros del clúster; por otro lado, el administrador de mensajería está optimizado para minimizar la cantidad de información necesaria para mantener actualizado el estado del sistema facilitando así el trabajo del administrador de membresía, ya que en lugar de enviar llamadas a funciones se envían mensajes que ayudan en la toma de decisiones.

### **3.5. INSTALACIÓN DEL ADMINISTRADOR DE RECURSOS.**

La quinta actividad consiste en instalar un administrador de recursos. El Administrador de recursos es el encargado de monitorear los recursos activos en el clúster de tal manera que estos puedan ser recuperados o transferidos automáticamente de un nodo a otro si se presenta un fallo. Por otro lado, este administrador se encarga de replicar automáticamente la configuración de ordenamiento y colocación de los nodos en el clúster.

---

<sup>3</sup> Véase Glosario - Encriptación

<sup>4</sup> Véase Glosario - Autenticación

### **3.6. INSTALACIÓN DE LOS SERVICIOS y APLICACIONES.**

La sexta actividad consiste en instalar las aplicaciones y los servicios. Desde el punto de vista de un administrador de red, el propósito principal de un clúster de alta disponibilidad es administrar los servicios y aplicaciones de los usuarios, sin embargo desde un punto vista más técnico y formal el clúster sirve para iniciar y detener recursos. Por ejemplo un servidor web dentro del clúster es simplemente un recurso.

## CAPITULO IV

### 4 . IMPLEMENTACION DEL CLÚSTER

En este prototipo de ejemplo se va a implementar un clúster de alta disponibilidad de 3 nodos, sin embargo, para demostrar que la estrategia planteada en el capítulo III puede ser aplicada para un clúster de alta disponibilidad de N nodos el diseño inicial del un clúster tendrá únicamente 2 nodos con los servicios de DNS y FTP para luego agregar un tercer nodo al clúster con el servicio de HTTP.

#### 4.1. DISEÑO DEL CLÚSTER CON 2 NODOS

La figura 7 muestra el diseño inicial del clúster, con los nodos A y B.

- Cualquier nodo puede proveer los recursos de DNS y FTP.
- Los recursos de DNS y FTP siempre deben estar en el mismo nodo.

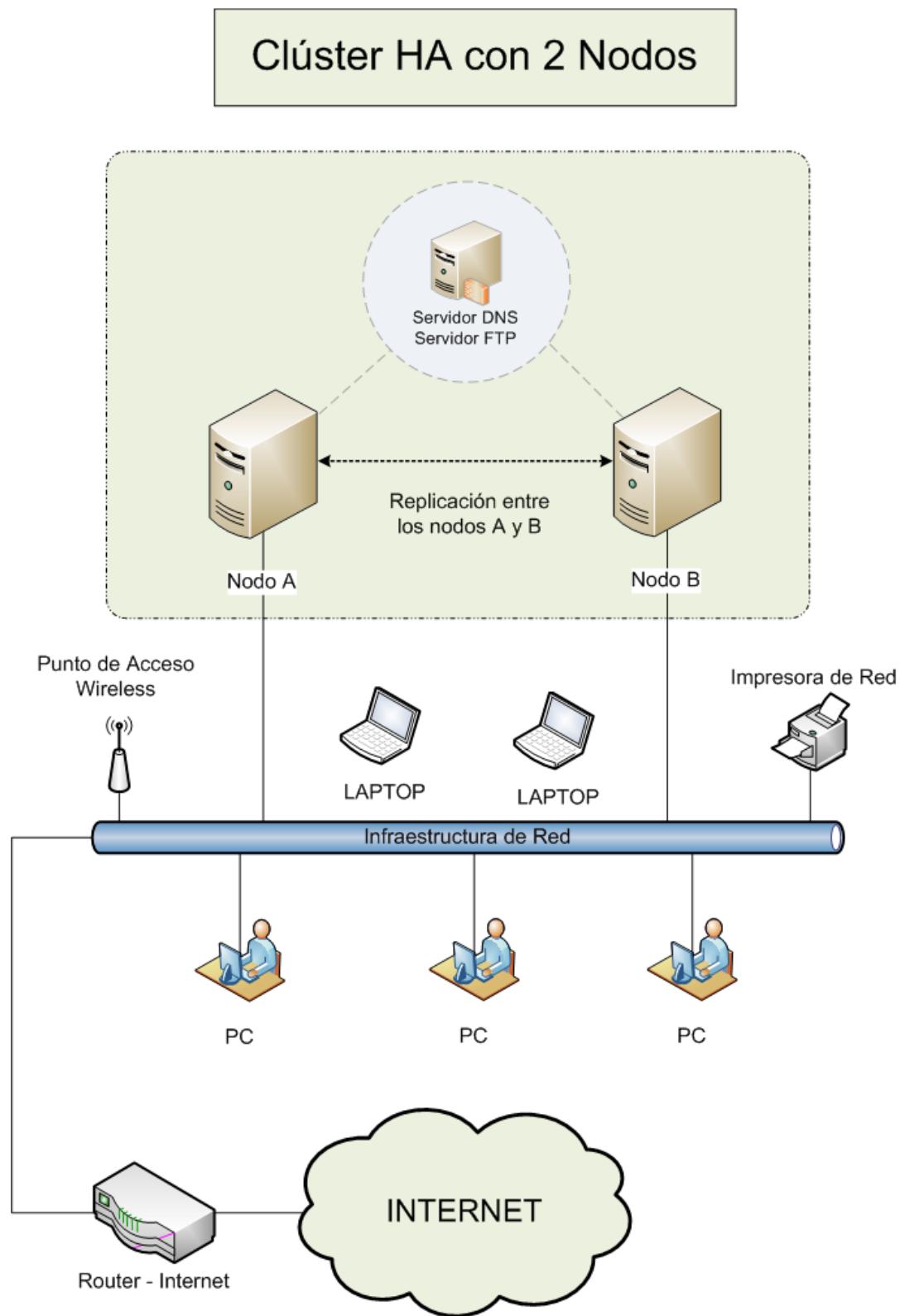


Figura 7. Diseño del clúster con 2 nodos.

#### 4.2. REQUERIMIENTOS DE HARDWARE

Tomando como punto de partida el escenario de ejemplo, el hardware mínimo necesario para implementar un clúster de nodos **dos** nodos es:

- 2 Computadores.
- Una conexión de red para comunicación TCP/IP.
- 2 unidades de almacenamiento para cada computador.
- Acceso a Internet.

#### 4.3. REQUERIMIENTOS DE SOFTWARE

La figura 8 muestra el software que vamos a instalar:

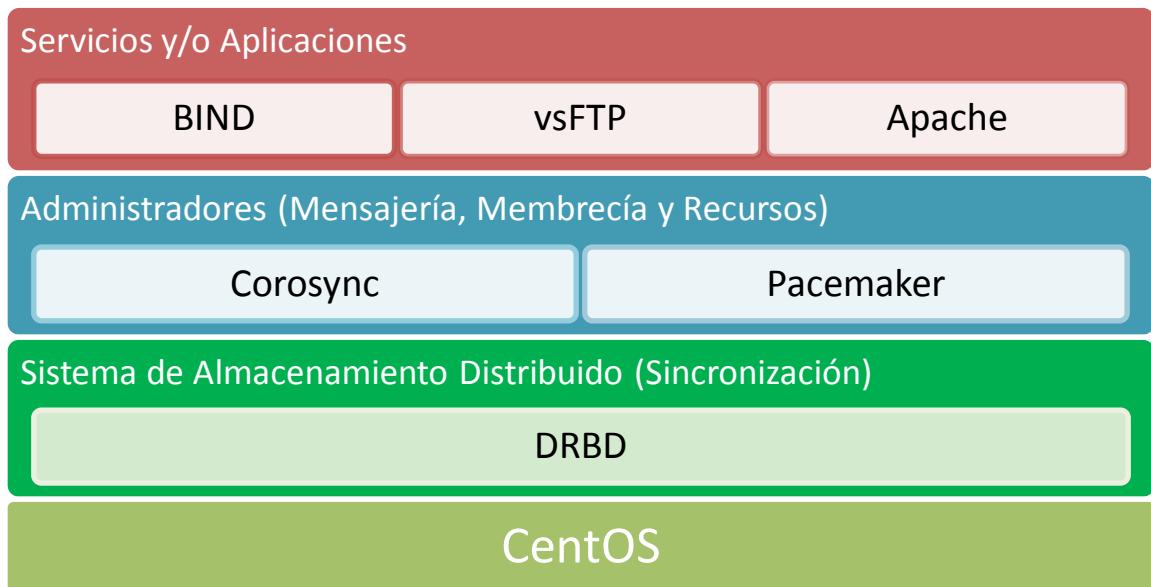


Figura 8. – Componentes de Software.

El software necesario para implementar el clúster es:

- Linux CentOS como Sistema Operativo.
- Pacemaker como Administrador de Recursos.
- Corosync como Administrador de Mensajería y Membrecía.
- DRBD como Administrador de Sincronización de datos.
- BIND y vsFTP como Recursos o Servicios.

#### **4.4. INSTALACION Y CONFIGURACIÓN DEL SISTEMA**

##### **OPERATIVO CentOS**

El Sistema Operativo seleccionado para esta implementación es Linux CentOS debido a que es un SO para servidores y prácticamente un clon de la distribución Red Hat Linux Enterprise ó RHLE, compilado a partir del código fuente liberado públicamente por Red Hat el cual no necesita de suscripción alguna para ser descargado desde el Internet.

Aunque CentOS no es mantenido ni asistido de ninguna manera por Red Hat, su similitud con RHLE hace que muchas empresas alrededor del mundo opten por esta distribución, por otro lado CentOS posee una comunidad muy grande de

desarrolladores voluntarios que colaboran con el soporte y actualización del software a través de foros, canales de chat IRC, listas de correo, FAQ y manuales en línea.

Algunas de las características que hacen de CentOS un SO muy popular son:

- Fácil mantenimiento.
- Desarrollo activo.
- Idóneo para entornos de producción a largo plazo.
- Infraestructura comunitaria.
- Modelo de gestión abierto.
- Soporte Comercial – ofrecido por un Socio Proveedor.

El DVD de Instalación de Linux CentOS puede ser descargado libremente desde alguno de los servidores listados en la siguiente página web:

<http://www.centos.org/modules/tinycontent/index.php?id=30>

Todos los pasos necesarios para instalar y configurar CentOS se encuentran detallados en el Anexo 1.

## 4.5. PREPARATIVOS

Durante el proceso de instalación es necesario definir, entre otras cosas, la estructura de particiones del disco y los parámetros de configuración red.

### 4.5.1.1. ESTRUCTURA DE LAS PARTICIONES .

Para este prototipo vamos a dividir el disco en 4 particiones primarias.

- La primera partición será usada para el directorio /boot que contiene el núcleo del sistema.
- La segunda partición será usada para la asignación de la memoria virtual o swap.
- La tercera partición será usada para el directorio raíz que contendrá el directorio del administrador
  - /**root**, el directorio /**home** de los usuarios, el directorio /**etc**, entre otros.
- La cuarta y última partición será usada para el directorio /var que contiene entre otras cosas los

archivos variables de registro o log, los archivos temporales de correo, etc.

#### **4.5.1.2. CONFIGURACIÓN DE RED.**

Para que todos los nodos dentro del clúster puedan comunicarse entre sí es necesario que todos pertenezcan a la misma red TCP/IP.

### **4.6. CONFIGURACIONES ADICIONALES**

#### **4.6.1.1. INSTALACIÓN DE LAS ACTUALIZACIONES DEL SISTEMA OPERATIVO**

La instalación de todas las actualizaciones del sistema disponibles se las puede realizar usando yum.

#### **4.6.1.2. CONFIGURACIÓN DEL ARCHIVO HOSTS**

Un requerimiento importante para que el clúster funcione correctamente es que todos los nodos deben ser capaces de comunicarse entre sí por el nombre de host, para ello es necesario editar el archivo hosts y añadir la siguiente información.

#### **4.6.1.3. CONFIGURACIÓN DEL REPOSITORIO DE RED**

##### **HAT**

La manera más sencilla de instalar el software para el clúster es a través de archivos ejecutables rpm, para ello es necesario agregar el repositorio público y oficial de Red Hat.

### **4.7. INSTALACIÓN DEL SISTEMA DE ALMACENAMIENTO**

#### **DISTRIBUIDO - DRBD**

El sistema de almacenamiento distribuido seleccionado para este prototipo es DRBD (Distributed Replicated Block Device) porque ha sido diseñado para construir clústeres de alta disponibilidad y permite replicas de múltiples dispositivos a través de una conexión de red.

Algunas de las características principales de DRBD son:

- Sincronización en tiempo real.
- Soporta discos de hasta 1 Petabyte ( $10^{15}$  bytes).
- Detección automática de bloques desactualizados.

- Re sincronización eficiente, sólo los bloques que fueron modificados son re sincronizados.
- Soporta TCP/IP sobre Ethernet.
- Permite los modos de operación síncrono y asíncrono.

DRBD soporta prácticamente cualquier dispositivo soportado por el kernel de Linux.

- Un disco completo.
- Una partición.
- Un administrador de volumen lógico.
- Un sistema de administración de volumen corporativo.

Todos los pasos necesarios para instalar y configurar DRBD se encuentran detallados en el Anexo 2.

#### **4.7.1. VISIÓN GENERAL DE DRBD**

DRBD es un software de replicación de datos que trabaja con la filosofía de “no compartir nada”. El objetivo de DRBD consiste en replicar los datos desde el servidor principal al servidor secundario de tal manera que los 2 servidores tengan la misma información. Esta replica se

realiza continuamente y puede ser vista como un RAID1 pero a través de la infraestructura de red.

Por defecto, DRBD usa el puerto TCP 7788 para la comunicación con otros nodos por lo tanto es necesario asegurarse que este puerto no esté bloqueado por ningún firewall. Por otro lado, DRBD reserva y oculta 128MB para los metadatos, es decir que si un dispositivo tiene 1GB de almacenamiento sólo estarán disponibles 896MB para los datos de las aplicaciones o recursos.

#### 4.7.1.1. COMPONENTES IMPORTANTES DE DRBD

La figura 9 muestra los componentes más importantes de la definición de un recurso DRBD.

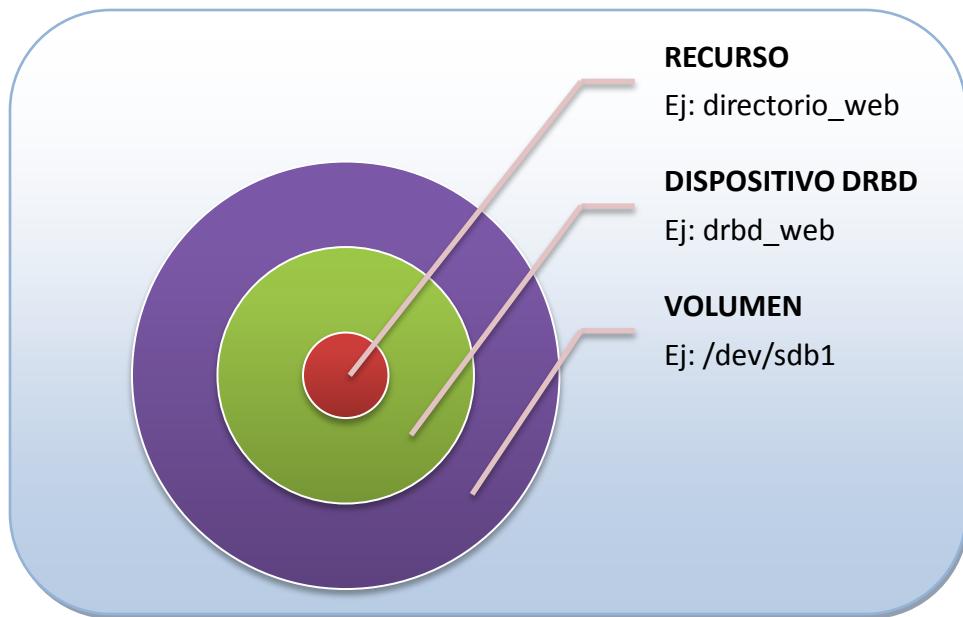


Figura 9. Componentes importantes de un recurso DRBD.

#### **4.7.1.2. RECURSO**

Es el término empleado para referirse a un conjunto determinado de datos replicados.

#### **4.7.1.3. DISPOSITIVO DRBD**

Es un dispositivo virtual de bloques administrado por DRBD. A cada dispositivo DRBD usualmente llamado drbdX, donde X representa un número, le corresponde un volumen en un recurso. DRBD también permite nombres personalizados pero estos deben iniciar con drbd\_.

#### **4.7.1.4. VOLUMEN**

Es lo que contiene los datos replicados y los metadatos de DRBD.

#### **4.7.1.5. CONEXIÓN**

Es un enlace de comunicación entre 2 equipos que comparten un conjunto de datos replicados.

### **4.7.2. ROLES**

A diferencia del concepto de activo y pasivo usado para las aplicaciones, el de concepto de primario y secundario

se usa para almacenamiento. Cada recurso puede tener un rol primario o secundario, cuando el estado del rol cambia de secundario a primario se conoce como promoción y cuando cambia de primario a secundario se conoce como degradación.

Un dispositivo BDRB en rol primario puede realizar tareas de lectura y escritura sin restricciones, sin embargo cuando el dispositivo tiene un rol secundario no puede ser usado en las tareas de lectura y escritura por las aplicaciones pues está restringido a recibir únicamente las actualizaciones desde el dispositivo en rol primario.

#### **4.7.3. CONVENCIONES GENERALES DE DRBD**

- El nombre de un recurso no puede contener espacios en blanco ni caracteres especiales.
- Los volúmenes se enumeran desde 0 hasta 65535.
- Los nombres de los dispositivos DRBD deben iniciar con la palabra drbd.

Los pasos para instalar y configurar DRBD son:

1. Instalar DBRD.

2. Editar el archivo /etc/drbd.conf.
3. Editar el archivo de configuración global.

#### **4.7.4. INSTALACIÓN DE DRBD**

Para instalar el sistema de almacenamiento distribuido desde el repositorio de Red Hat vamos a usar yum.

#### **4.7.5. CONFIGURACIÓN INICIAL DE DRBD**

La configuración inicial de drbd consiste en definir la ubicación de los archivos de configuración global y la de los recursos.

### **4.8. INSTALACIÓN DEL ADMINISTRADOR DE MEMBRECÍA Y MENSAJERÍA – COROSYNC**

El administrador de membrecía y mensajería seleccionado es Corosync, dado que es una versión derivada de OpenAIS y está respaldado tanto por Red Hat como por Novell/Suse. Por otro lado contiene todos protocolos de red necesarios para la intercomunicación de los procesos y para la implementación general del clúster HA sin que tenga dependencia con alguna distribución específica de Linux.

Algunas de las características principales de Corosync son:

- La tecnología usada por Corosync se basa en más de 20 años de investigación en computación distribuida.
- Soporte para encriptación y autenticación para una comunicación segura.
- Soportado por la mayoría de distribuciones Linux.
- Diseño compacto con menos de 60 mil líneas de código.
- Diseñado optimizado para minimizar las copias a memoria y los intercambios de contexto.
- Alto rendimiento en redes Ethernet y 10G InfiniBand.
- Disponible para Linux, Solaris, BSD, Darwin.

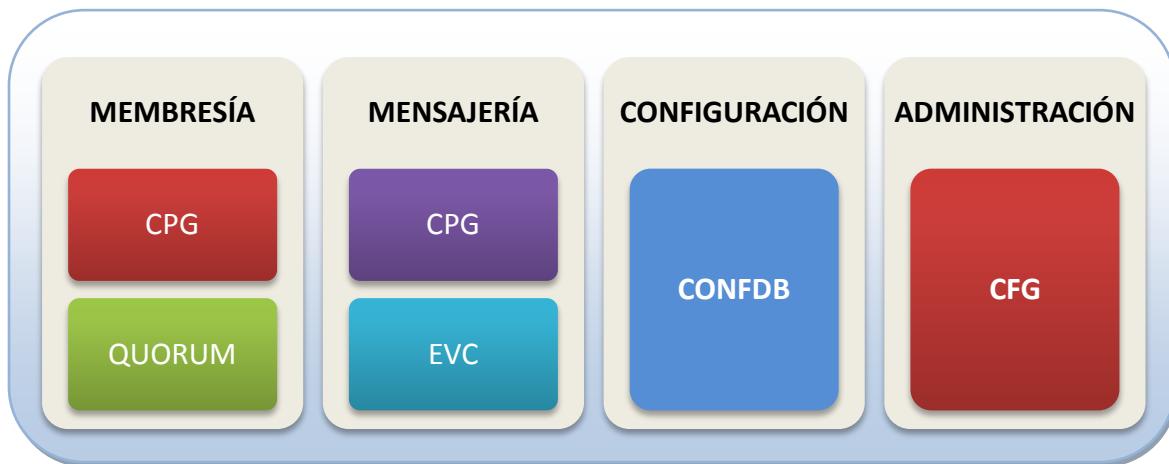
Corosync aborda la alta disponibilidad asegurando que cada servidor redundante en el sistema mantiene una copia redundante de información que se usa para tomar decisiones acerca de las aplicaciones. Por otro lado, en lugar de enviar llamadas a funciones como se hace en un enfoque de

estado de máquina típico, Corosync envía estados de máquina a todos los nodos del clúster de manera ordenada y consistente.

Todos los pasos necesarios para instalar y configurar Corosync se encuentran detallados en el Anexo 3.

#### **4.8.1. COMPONENTES PRINCIPALES DE COROSYNC**

La figura 10 muestra los componentes principales de Corosync.



**Figura 10. Componentes principales de Corosync.**

##### **4.8.1.1. GRUPO DE PROCESO CERRADO - CPG**

Provee membresía dentro de las aplicaciones. Cuando una aplicación se une a un proceso de grupo, todas las aplicaciones envían un mensaje indicando un cambio de

membrecía que incluye el ID del proceso de la aplicación y el ID del nodo. Cuando una aplicación es parte de un grupo, esta puede enviar y recibir mensajes a todos los miembros del grupo.

#### **4.8.1.2. QUORUM**

Provee información útil para determinar si el clúster tiene quórum en el caso que la mayoría de nodos del clúster está presente.

#### **4.8.1.3. SINCRONÍA VIRTUAL EXTENDIDA - DEVS**

Es la interfaz extendida virtual de mensajería para sincronización entre nodos.

#### **4.8.1.4. CONFDB**

Provee acceso a la configuración y estadísticas.

#### **4.8.1.5. CFG**

Es la API de configuración que provee los mecanismos para el control del estado de operación de Corosync.

## 4.9. INSTALACIÓN DEL ADMINISTRADOR DE RECURSOS -

### PACEMAKER

El administrador de recursos seleccionado es Pacemaker, pues no es dependiente de ninguna distribución en especial y se integra perfectamente con Corosync.

Algunas de las características principales de Pacemaker son:

- Detección y recuperación nodos y errores a nivel de servicio.
- Independiente del tipo de almacenamiento agnóstico, es decir que no requiere almacenamiento compartido.
- Independiente de los Recursos, es decir que cualquier cosa que pueda ser puesta en un archivo de secuencia de comandos puede ser parte del clúster.
- Soporta Clústeres grandes o pequeños.
- Prácticamente soporta cualquier configuración de redundancia.
- La configuración puede ser actualizada en cualquier nodo y es automáticamente replicada a todos los nodos.

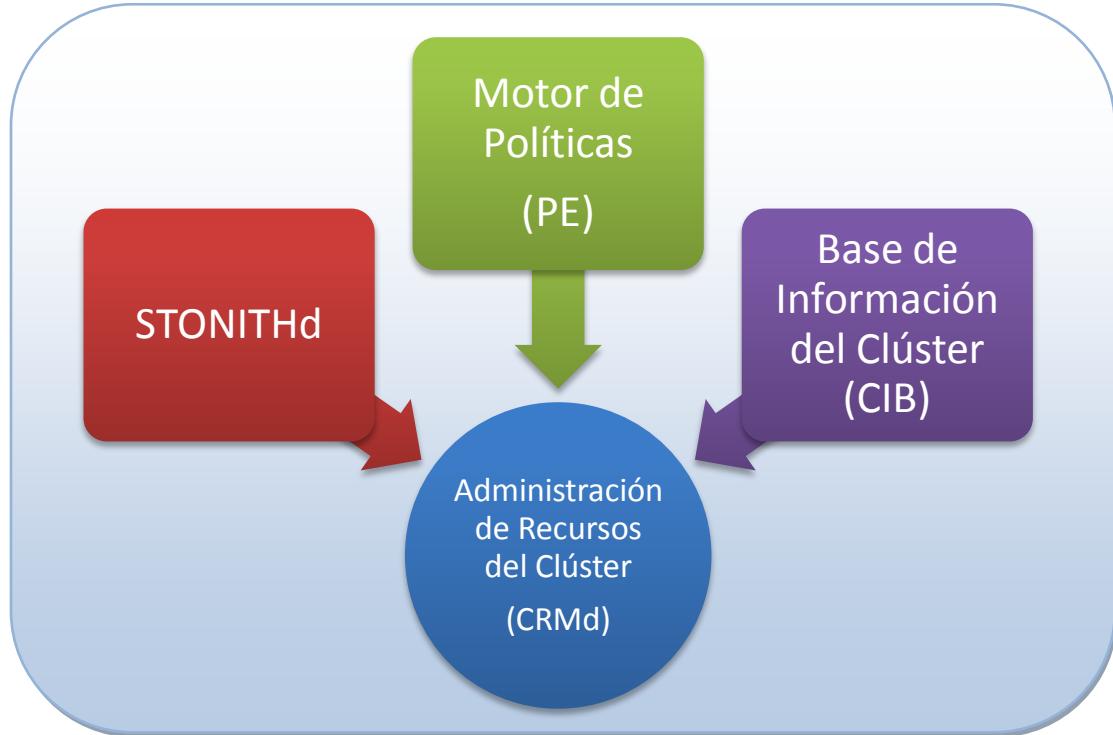
Todos los pasos necesarios para instalar y configurar Pacemaker se encuentran detallados en el Anexo 4.

#### **4.9.1. VISION GENERAL DE PACEMAKER**

Pacemaker centraliza todas las decisiones del clúster nombrando a un miembro como coordinador designado. Este coordinador se encarga de enviar de pasar las políticas del clúster a los administradores de recursos locales en cada nodo. Si el nodo coordinador falla, un nuevo coordinador es elegido. Por otro lado, si existe algún cambio de estado, todos los nodos tienen que reportar los resultados de sus operaciones al coordinador designado para que se evalúe el estado del clúster y determinar si es necesario tomar nuevas acciones.

#### **4.9.1. COMPONENTES INTERNOS DE PACEMAKER**

La figura 11 muestra los componentes internos de Pacemaker.



**Figura 11. Componentes internos de Pacemaker.**

#### **4.9.1.1. BASE DE INFORMACIÓN DEL CLÚSTER**

Contiene todas las definiciones de las opciones de configuración del clúster, los nodos, los recursos y las relaciones entre nodos con sus respectivos estados.

#### **4.9.1.2. MOTOR DE POLÍTICAS**

Es el encargado de calcular el próximo estado del clúster basado en el estado y la configuración actual.

#### **4.9.1.3. STONITH**

Es un mecanismo de aislamiento para los nodos con problemas con la finalidad de prevenir corrupción de datos.

Estos componentes se integran de la siguiente manera:

La Base de Información de Clúster (CIB) utiliza lenguaje XML para representar la configuración y el estado actual de todos los recursos del clúster. El contenido del CIB se mantiene automáticamente sincronizado en todo el clúster y es usado por el Motor de Políticas (PE) para calcular el estado ideal del clúster y cómo debe ser alcanzado.

#### **4.9.2. AGENTES**

Los agentes son las interfaces encargadas de abstraer los recursos dentro del clúster, es decir cada agente se encarga de reaccionar adecuadamente cuando hay que iniciar o detener un recurso a través de archivos de secuencia de comandos.

Pacemaker soporta los siguientes tipos de agentes:

- Agentes heredados de Hearbeat 1
- Archivos de secuencia de comandos basados en estándares de Linux. (LSB)
- Agentes basados en Framework de Clúster Abierto (OCF)

#### **4.9.3. TIPO DE RECURSOS**

##### **4.9.3.1. PRIMITIVO**

Es el tipo de recurso más básico dentro del clúster, pues no depende de otros recursos para ser iniciado.

##### **4.9.3.2. GRUPO**

Es un conjunto de recursos que dependen de otros recursos y necesitan iniciarse en un orden específico dentro de un mismo nodo.

##### **4.9.3.3. CLON**

Es aquel tipo de recurso que puede estar activo en múltiples nodos al mismo tiempo.

#### **4.9.3.4. MÁSTER**

Es un tipo especial de clon que permite dos modos de operación: máster y esclavo.

### **4.9.4. RESTRICCIONES DE LOS RECURSOS**

Existen tres diferentes tipos de restricciones disponibles en Pacemaker:

#### **4.9.4.1. UBICACIÓN**

Las restricciones de ubicación definen los nodos en que un recurso puede ser ejecutado, no puede ser ejecutado o es preferible que sea ejecutado.

#### **4.9.4.2. CO-UBICACIÓN**

Las restricciones de co-ubicación definen los recursos que se pueden ejecutar y que no se pueden ejecutar en un mismo nodo.

#### **4.9.4.3. ORDEN**

Las restricciones de orden definen la secuencia de ejecución de las acciones sobre los recursos.

#### 4.9.5. PUNTUACIONES E INFINITO

Para poder definir las restricciones correctamente es necesario definir las puntuaciones, pues cada restricción debe ir acompañada de una puntuación. En términos generales las puntuaciones son muy importantes porque ayudan a decidir cuando un recurso debe ser detenido o migrado. Hay que tomar en cuenta que todas las puntuaciones son calculadas por recurso y las restricciones con puntuación más alta se aplican primero, cuando un recurso necesita ser migrado el clúster escoge el nodo con la puntuación más alta como nuevo anfitrión, sin embargo, ningún nodo con una puntuación negativa puede ser anfitrión de ese recurso.

El valor de infinito se define como 1,000.000; cualquier operación de suma o resta del infinito debe seguir las siguientes reglas:

- $X + \text{INFINITO} = \text{INFINITO}$
- $X - \text{INFINITO} = -\text{INFINITO}$
- $\text{INFINITO} - \text{INFINITO} = -\text{INFINITO}$

#### 4.9.6. NODOS DE RECUPERACIÓN

Un recurso migrado puede ser restaurado a su nodo original cuando ha superado el problema que originó la migración. Para prevenir este comportamiento o definir un nuevo nodo anfitrión es necesario cambiar el valor de la adherencia<sup>5</sup> del recurso.

Hay que tomar en cuenta las siguientes implicaciones al momento de cambiar el valor de adherencia:

- **IGUAL A 0:** El recurso migrará al nodo original automáticamente a menos que se haya movido a un nodo distinto al original.
- **MAYOR A 0:** El recurso permanecerá en el nodo actual pero puede ser migrado si otro nodo más apropiado está disponible.
- **MENOR A 0:** El recurso será migrado de su ubicación actual cuando otro nodo más apropiado esté disponible.

---

<sup>5</sup>Véase Glosario - Valor de Adherencia

- **INFINITO:** El recurso siempre permanecerá en la ubicación actual mientras el nodo esté disponible. Esta opción prácticamente deshabilita la recuperación automática.
- **-INFINITO:** El recurso siempre se moverá de la ubicación actual.

## **4.10. INSTALACIÓN DE LOS RECURSOS**

### **4.10.1. INSTALACIÓN DEL SERVIDOR PARA RESOLUCIÓN DE NOMBRES DE DOMINIO**

El servidor seleccionado para este proyecto es BIND, que será el encargado de traducir los nombres de dominio a direcciones IP dentro del Clúster.

Todos los pasos necesarios para instalar y configurar el servidor DNS se encuentran detallados en el Anexo 6.

#### **4.10.1.1. SERVIDOR MAESTRO**

Atiende las peticiones de resolución de nombres directamente dentro de la red local y puede reenviar las peticiones a servidores externos en caso de ser necesario.

#### **4.10.1.2. ZONA MAESTRA**

Traduce los nombres de dominio a direcciones IP.

#### **4.10.1.3. ZONA INVERSA**

Traduce las direcciones IP a nombres de dominio.

### **4.10.2. INSTALACIÓN DEL SERVIDOR DE TRANSFERENCIA DE ARCHIVOS**

El servidor seleccionado para este proyecto es vsFTPD, que será el encargado alojar los archivos en el servidor.

Todos los pasos necesarios para instalar y configurar el servidor FTP se encuentran detallados en el Anexo 7.

#### **4.10.2.1. SERVIDOR FTP**

Es el programa que controla el acceso a los archivos y responde las peticiones de los clientes FTP.

#### **4.10.2.2. CLIENTE FTP**

Es el programa que permite enviar o recibir archivos al servidor FTP.

#### **4.11. AGREGAR UN TERCER NODO AL CLÚSTER ACTUAL**

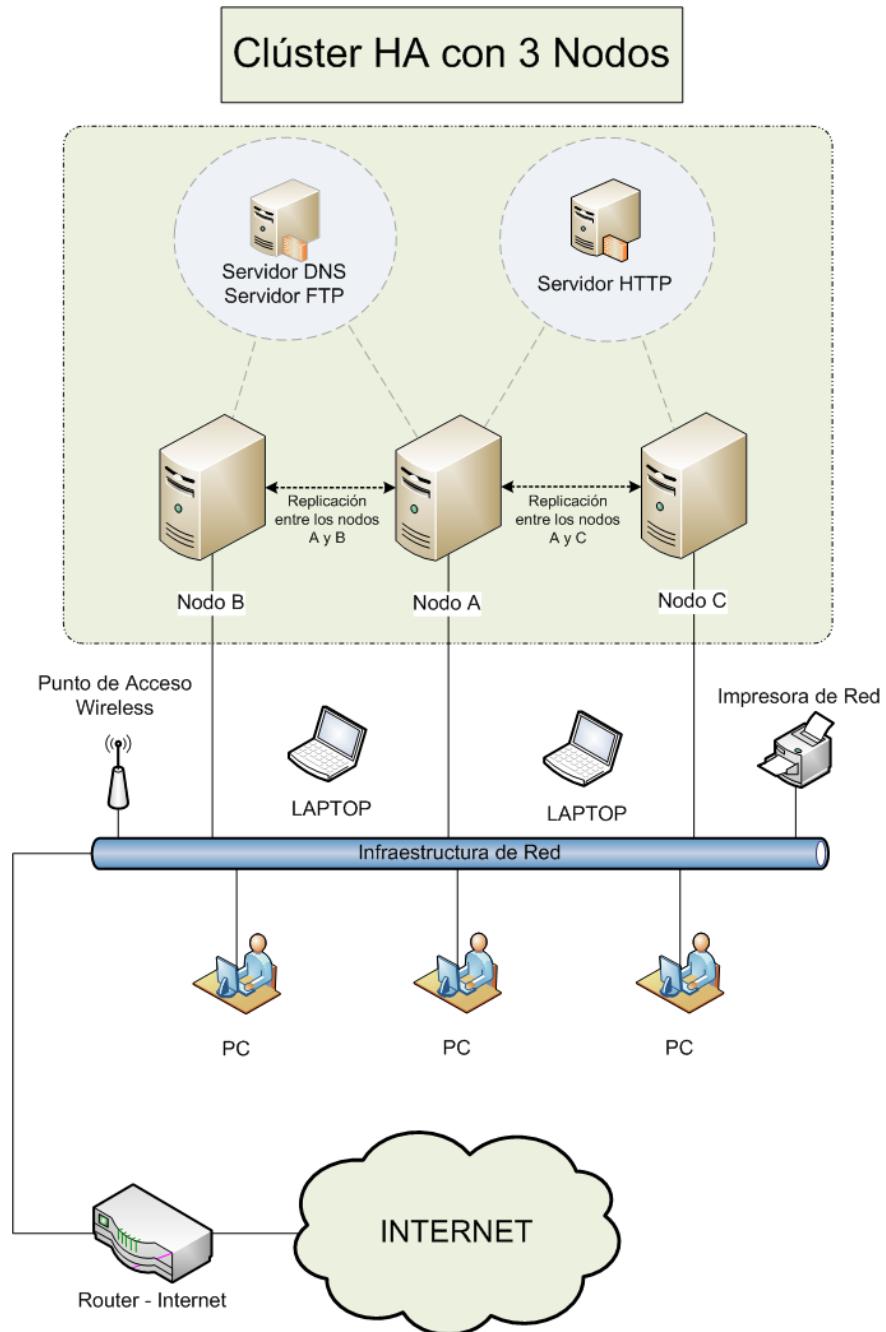
Para agregar el tercer nodo a la arquitectura actual del clúster es necesario repetir las mismas actividades realizadas para los dos primeros nodos, empezando por el diseño del nuevo clúster hasta llegar a las pruebas funcionales.

##### **4.11.1. DISEÑO DEL CLÚSTER CON 3 NODOS**

El diseño del clúster con 3 nodos debe ajustarse a las siguientes características.

- El nodo A ó el Nodo B pueden ofrecer los servicios de DNS y FTP.
- El nodo A ó el Nodo C pueden ofrecer el servicio de HTTP.
- El nodo A puede ofrecer los servicios de HTTP, DNS ó FTP.
- El nodo B NO puede ofrecer el servicio de HTTP.
- El nodo C NO puede ofrecer el servicio de DNS ó FTP.

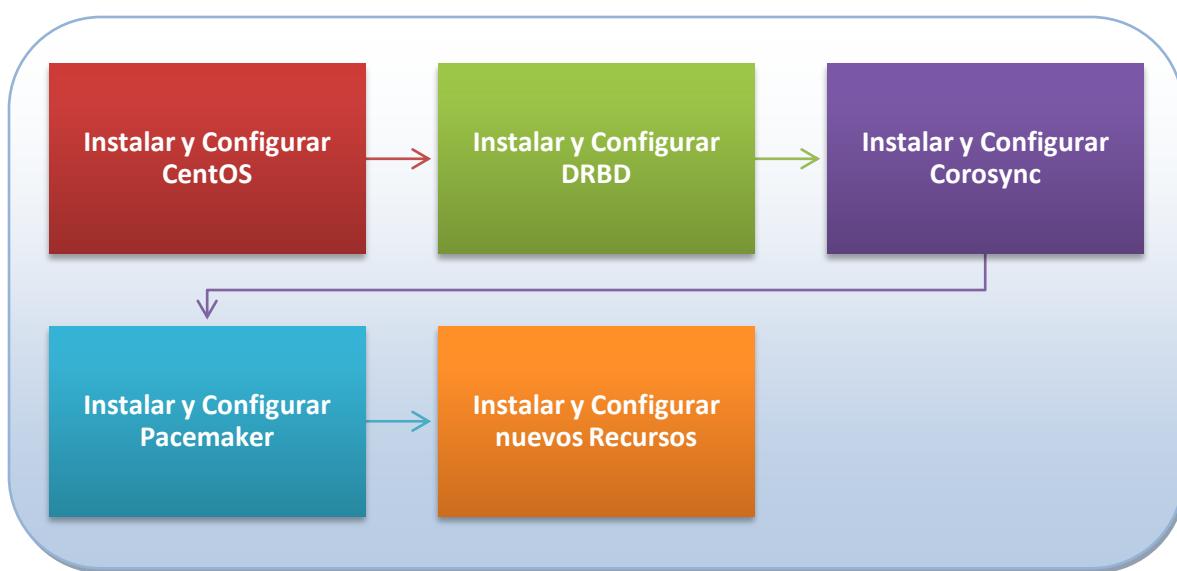
Tomando en cuenta las nuevas características del clúster, el diseño final del clúster con tres nodos se puede ver en la figura 12.



**Figura 12. Diseño del clúster con 3 nodos.**

#### 4.11.2. CONFIGURACIÓN DEL CLÚSTER CON 3 NODOS

Para que la configuración de clúster se adapte al nuevo diseño es necesario realizar las actividades que se muestran en la figura 13.



**Figura 13. Actividades de Configuración en el Tercer Nodo.**

El detalle de los cambios de configuración se encuentra detallado en el Anexo 9.

Para terminar la configuración del tercer nodo, es necesario agregar un recurso adicional al clúster de tal manera que se ajuste al diseño definido en la figura 12.

#### **4.11.3. INSTALACIÓN DEL SERVIDOR WEB**

El servidor Web seleccionado para este proyecto es Apache, que será el encargado de recibir y responder a las peticiones Web que se realicen dentro del Clúster.

Todos los pasos necesarios para instalar y configurar el servidor HTTP se encuentran detallados en el Anexo 11.

##### **4.11.3.1. PÁGINA WEB**

Es el nombre con el que se conoce a los documentos adaptados para ser visualizados en un navegador Web.

#### **4.12. ESTRATEGIA PARA AGREGAR NODOS ADICIONALES AL CLÚSTER**

Para agregar un nuevo nodo al clúster, es necesario realizar las actividades que se muestran en la figura 14.

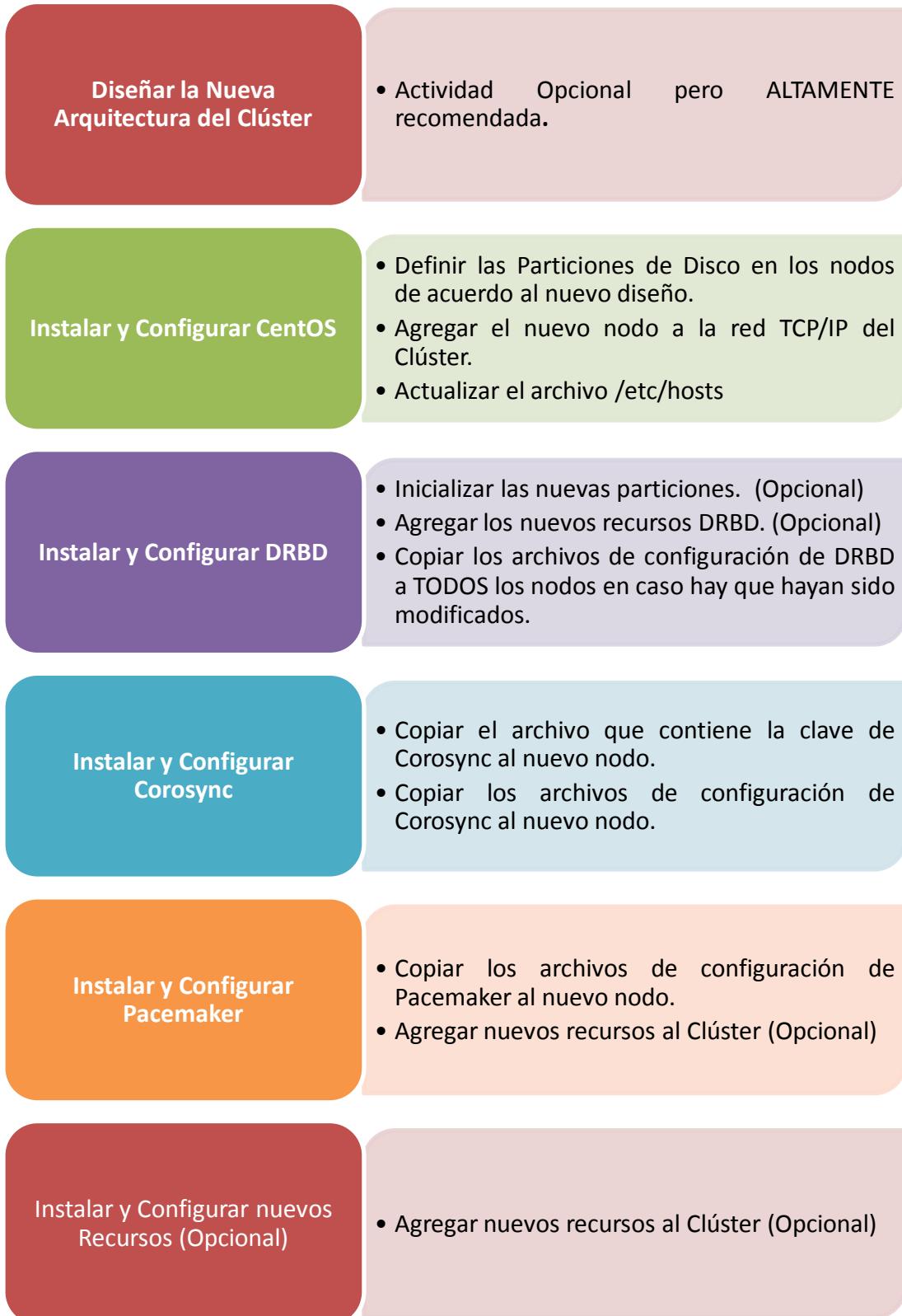


Figura 14. Actividades para agregar un nodo adicional al clúster.

#### **4.13. COMANDOS PRINCIPALES PARA LA ADMINISTRACIÓN DEL CLÚSTER**

Para ver una lista de los comandos más útiles de administración del clúster con una breve explicación de los mismos, véase el anexo 11.

## CAPITULO V

### 5. PRUEBAS FUNCIONALES Y RESULTADOS

A diferencia de las pruebas individuales realizadas durante la fase de configuración de los componentes del clúster, las pruebas funcionales consisten en verificar que todos los componentes dentro del clúster funcionen correctamente, tanto en las tareas de migración como de replicación.

#### 5.1. PRUEBAS DE REPLICACIÓN DEL CLÚSTER CON 2

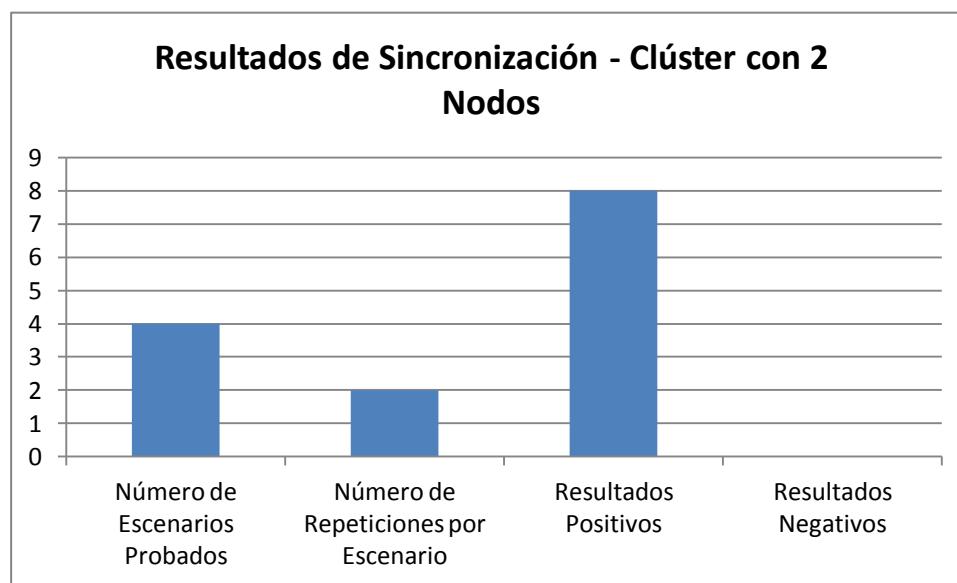
##### NODOS

De acuerdo a la figura 7, la replicación en el clúster debe hacerse entre los datos de los recursos del Nodo A y del nodo B, por lo tanto vamos a verificar que los cambios realizados en los archivos ó carpetas de los recursos de un nodo se repliquen al otro nodo.

En la tabla 1 se muestran los escenarios de prueba para el proceso de replicación con sus resultados. Como se puede observar, todas las pruebas se realizaron correctamente y los resultados fueron positivos.

ESTADO ACTUAL DEL CLÚSTER	PRUEBA	RESULTADO ESPERADO	RESULTADO DE LA PRUEBA
El nodo A provee los recursos de DNS y FTP.	Un nuevo archivo es creado en la carpeta del servidor FTP en el nodo A.	El Archivo es copiado al Nodo B	✓ OK
El nodo A provee los recursos de DNS y FTP.	Una nueva dirección es registrada en el servidor DNS del nodo A.	El registro es actualizado en el Nodo B	✓ OK
El nodo B provee los recursos de DNS y FTP.	Un nuevo archivo es creado en la carpeta del servidor FTP en el nodo B.	El Archivo es copiado al Nodo A	✓ OK
El nodo B provee los recursos de DNS y FTP.	Una nueva dirección es registrada en el servidor DNS del nodo B.	El registro es actualizado en el Nodo A	✓ OK

**Tabla 1.** Escenarios para pruebas de replicación - Clúster con 2 nodos.



**Figura 15.** Resultados de Sincronización - Clúster con 2 nodos.

En la tabla 2 se muestran los resultados de las pruebas de replicación de archivos desde el nodo A hacia al nodo B.

ESTADO ACTUAL DEL CLÚSTER	TAMAÑO DEL ARCHIVO (MB)	TIEMPO DE REPLICACIÓN (SEGUNDOS)
El nodo A provee los recursos de DNS y FTP.	5.1 MB	1.8 seg.
El nodo A provee los recursos de DNS y FTP.	9.9 MB	3.9 seg.
El nodo A provee los recursos de DNS y FTP.	20.2 MB	8 seg.
El nodo A provee los recursos de DNS y FTP.	40.3 MB	16.5 seg.
El nodo A provee los recursos de DNS y FTP.	79.9 MB	31.1 seg.
El nodo A provee los recursos de DNS y FTP.	159.6 MB	64.2 seg.
El nodo A provee los recursos de DNS y FTP.	320.3 MB	128.6 seg.

**Tabla 2. Pruebas de Replicación de Archivos - Clúster con 2 nodos**

Tomando en cuenta los resultados de estas pruebas se puede ver que la estrategia de implementación propuesta para el

proceso de replicación funciona correctamente y DRBD está correctamente integrado dentro del clúster con 2 nodos.

### **5.2. PRUEBAS DE MIGRACIÓN DEL CLÚSTER CON 2 NODOS**

De acuerdo a la figura 7, la migración de recursos debe hacerse entre los recursos del Nodo A y del nodo B, por lo tanto es necesario verificar que los recursos de un nodo se transfieran al otro. Los escenarios de migración evaluados en el clúster con 2 nodos se encuentran detallados en la siguiente tabla.

ESTADO ACTUAL DEL CLÚSTER	PRUEBA	RESULTADO ESPERADO	OK ERROR
El nodo A provee los recursos de DNS y FTP.	El nodo A es reiniciado.	Los recursos pasan al nodo B.	✓ OK
El Nodo B provee los recursos de DNS y FTP	El nodo B es reiniciado.	Los recursos pasan al nodo A.	✓ OK
El nodo A provee los recursos de DNS y FTP.	El nodo A pierde conexión de red.	Los recursos pasan al nodo B.	✓ OK
El Nodo B provee los recursos de DNS y FTP	El nodo B pierde conexión de red.	Los recursos pasan al nodo A.	✓ OK
El nodo A provee los recursos de DNS y FTP.	Los nodos A y B son reiniciados.	Los recursos no deben ser asignados a ningún nodo.	✓ OK
Los recursos no han sido asignados a ningún nodo.	Reiniciar el nodo B.	Los recursos pasan al nodo A.	✓ OK
Los recursos no han sido asignados a ningún nodo.	Reiniciar el nodo A.	Los recursos pasan al nodo B.	✓ OK

**Tabla 3. Pruebas de Migración de Recursos – Clúster con 2 nodos**

Tomando en cuenta los resultados de estas pruebas se puede ver que la estrategia de implementación propuesta para el proceso de migración funciona correctamente, por lo tanto Corosync y Pacemaker están correctamente integrados dentro del clúster con 2 nodos.

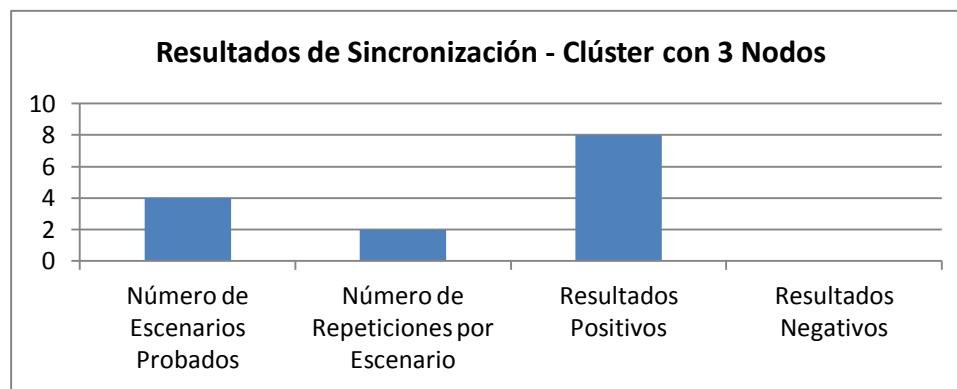
### **5.3. PRUEBAS FUNCIONALES DEL CLÚSTER CON 3 NODOS**

Las pruebas funcionales del clúster con 3 nodos son similares a las realizadas en el clúster con 2 nodos. De acuerdo a la figura 12, la replicación en el clúster debe hacerse entre los datos de los recursos del Nodo A y del nodo B, así como los datos de los recursos del Nodo A y del nodo C. Los escenarios de migración evaluados en el clúster con 3 nodos se encuentran detallados en la tabla 4.

<b>ESTADO ACTUAL DEL CLÚSTER</b>	<b>PRUEBA</b>	<b>RESULTADO ESPERADO</b>	<b>RESULTADO</b>
El nodo A provee los recursos de DNS, FTP y HTTP.	Un nuevo archivo es creado en la carpeta de servidor FTP del nodo A.	El Archivo es copiado al Nodo B	OK
El nodo A provee los recursos de DNS, FTP y HTTP.	Una nueva dirección es registrada en el servidor DNS del nodo A.	El registro es actualizado en el Nodo B	OK
El nodo B provee los recursos de DNS y FTP.	Un nuevo archivo es creado en la carpeta de servidor FTP del nodo B.	El Archivo es copiado al Nodo A	OK

El nodo B provee los recursos de DNS y FTP.	Una nueva dirección es registrada en el servidor DNS del nodo B.	El registro es actualizado en el Nodo A	OK
El nodo A provee los recursos de DNS, FTP y HTTP.	Un nuevo archivo es creado en la carpeta de servidor HTTP del nodo A.	El Archivo es copiado al Nodo C	OK
El nodo A provee el recurso de HTTP.	Un nuevo archivo es creado en la carpeta de servidor HTTP del nodo C.	El Archivo es copiado al Nodo A	OK

**Tabla 4. Escenarios para pruebas de replicación - Clúster con 3 nodos.**



**Figura 16. Resultados de Sincronización - Clúster con 3 nodos.**

#### 5.4. PRUEBAS DE MIGRACIÓN DEL CLÚSTER CON 3 NODOS

De acuerdo a la figura 12, la migración de recursos FTP y DNS debe hacerse entre los nodos A y B. Así mismo, la migración del recurso HTTP debe hacerse entre los nodos A y C. Los escenarios de migración evaluados en el clúster con 3 nodos se encuentran detallados en la siguiente tabla.

ESTADO ACTUAL DEL CLÚSTER	PRUEBA	RESULTADO ESPERADO	OK ERROR
El nodo A provee los recursos de DNS FTP y HTTP.	El nodo A es reiniciado.	Los recursos de DNS y FTP pasan al nodo B y el recurso HTTP pasa al nodo C.	OK
El nodo B provee los recursos de DNS y FTP. El Nodo C provee el recurso de HTTP.	El nodo B es reiniciado.	El recurso de HTTP pasa al nodo A.	OK
El nodo B provee los recursos de DNS y FTP. El Nodo C provee el recurso de HTTP.	El nodo C es reiniciado.	El recurso de HTTP pasa al nodo A.	OK
El nodo B provee los recursos de DNS y FTP. El Nodo C provee el recurso de HTTP.	Los nodos B y C son reiniciados.	Todos los recursos pasan al nodo A.	OK
El nodo A provee todos los recursos.	Los nodos A, B y C son reiniciados.	Los recursos no deben ser asignados a ningún nodo.	OK
Los recursos no han sido asignados a ningún nodo.	Reiniciar el nodo A.	Los recursos de DNS y FTP pasan al nodo B y el recurso HTTP pasa al nodo C.	OK

**Tabla 5. Pruebas de Migración de Recursos – Clúster con 3 nodos**

De acuerdo a los resultados obtenidos, se puede ver que la estrategia de implementación propuesta para los procesos de replicación y migración también funciona para el clúster con 3 nodos, por lo tanto los componentes principales del clúster como DRBD, Corosync, Pacemaker, así como los recursos de DNS, FTP y HTPTP están correctamente integrados dentro del clúster.

## 6. CONCLUSIONES Y RECOMENDACIONES

### 6.1. CONCLUSIONES

Con la realización de este trabajo se definió una estrategia para la implementación de un clúster de alta disponibilidad de N nodos usando software libre, que puede ser usada para proveer redundancia a las aplicaciones críticas de una empresa. A través de este trabajo, también fue posible proveer un sistema de tolerancia a fallas a servidores Linux usando un clúster de Alta Disponibilidad, cumpliendo de esta manera una de las metas planteadas al inicio del proyecto.

En el desarrollo teórico del proyecto se definieron los componentes básicos para la implementación de un clúster HA y se explicaron los aspectos más importantes de cada uno, además se mencionó la función que tiene cada componente y como se integran dentro del clúster de Alta Disponibilidad.

Con la finalidad de demostrar que la estrategia planteada si funciona, primero se diseñó e implementó un clúster de alta disponibilidad con 2 nodos para ofrecer los servicios de FTP y DNS. Luego se añadió un nodo adicional para ofrecer el servicio de HTTP y se modificó el diseño original de tal

manera que los recursos sean distribuidos a nodos específicos dentro del clúster y que los servicios de FTP y DNS estén siempre juntos en un mismo nodo; logrando así un ejemplo de implementación para un escenario avanzado.

Durante la implementación del prototipo del proyecto también se indicó paso a paso cómo instalar, configurar, integrar y probar el software necesario para formar un clúster de alta disponibilidad. Así mismo se mostró cómo modificar la configuración del clúster para agregar un nodo adicional y se dio una breve explicación de los comandos más útiles para la administración del clúster.

Es importante mencionar que pese a que los problemas encontrados durante la instalación de los componentes para formar el clúster fueron solucionados probando diferentes versiones del software, los problemas encontrados durante la configuración e integración de cada componente demandaron más lectura e investigación que la esperada, sobre todo por problemas de incompatibilidad y errores de configuración ya que la información encontrada en los documentos oficiales, libros y artículos en Internet está desactualizada e incompleta, incluso fue necesario leer el código fuente de algunas aplicaciones.

Con los resultados obtenidos durante la realización de este trabajo puedo afirmar que Linux y el software de distribución libre son una alternativa real para la implementación de un clúster de alta disponibilidad. Una empresa podría usar la propuesta planteada en este trabajo y en lugar de pagar regalías por usar software propietario podría pagar la capacitación del personal técnico en el Sistema Operativo Linux obteniendo así un valor agregado a la implementación del proyecto.

Aunque los aspectos relacionados con la seguridad, escalabilidad, optimización o rendimiento del clúster están fuera del alcance de este proyecto, por no ser temas triviales, estos deben ser considerados para una implementación en un ambiente real. Tomando en cuenta lo dicho anteriormente, este trabajo puede servir como punto de partida para realizar profundizar esos temas en trabajos posteriores.

Como experiencia personal puedo decir que con este trabajo he podido enriquecer mi conocimiento acerca de los sistemas Linux y el software de libre distribución. Así mismo, he comprendido que para poder usar software libre en proyectos como este, es necesario estar dispuesto a enfrentarse a

problemas poco conocidos y en algunas ocasiones totalmente desconocidos, que para solucionarlos requieren de paciencia, dedicación y sobre todo el deseo de aprender a través de la investigación y autoeducación.

Como conclusión final y quizá las más importante a la que he podido llegar, es que la estrategia de implementación propuesta y el software sugerido en este trabajo permiten la implementación de clústeres de alta disponibilidad de N nodos con diseños complejos, como el presentado en la implementación del prototipo con 3 nodos y 3 servicios.

## 6.2. RECOMENDACIONES

A continuación se listan algunas de las recomendaciones más importantes para la implementación de un clúster HA:

- Verificar que el hardware que se va a usar esté soportado o certificado por el Sistema Operativo.
- Nunca usar versiones beta de software en un clúster HA en un ambiente de producción.
- Usar hardware del mismo proveedor en los nodos del clúster, especialmente las tarjetas de red y los medios de almacenamiento.
- Nunca actualizar el software de los componentes del clúster en un ambiente de producción sin haber realizado las pruebas respectivas.
- Revisar periódicamente los archivos de logs de los nodos del clúster.
- Revisar regularmente los anuncios oficiales del software que se esté usando, es posible que existan

actualizaciones para mejorar el software, solución de errores o parches de seguridad.

- Crear planes para realizar tareas de mantenimiento de hardware en los nodos.
- Documentar el diseño y la configuración del clúster.
- Familiarizarse con la administración de clúster antes de ponerlo en producción.

Por otro lado, para poder implementar satisfactoriamente un clúster HA es recomendable leer la documentación oficial del software que se va a utilizar, además es necesario entender los conceptos y definiciones básicas de los mismos. Por otro lado, nunca está de más buscar o pedir información de casos de éxito en proyectos similares que sirvan de referencia o punto de partida para la planeación del diseño, instalación, configuración o pruebas del software seleccionado, pero sobre todo para la correcta estimación de los recursos y alcance final del proyecto.

También hay que tomar en cuenta que aunque la implementación de un clúster HA puede ser una buena solución para los

problemas de falta de redundancia en las aplicaciones críticas de una empresa, también puede crear otros problemas y convertirse en un dolor de cabeza para los administradores de sistemas, especialmente si el hardware que soporta las aplicaciones no es fiable. Se recomienda seleccionar sabiamente el hardware que se va a usar en un clúster de alta disponibilidad.

En la actualidad, la alta disponibilidad de las aplicaciones es muy importante no solo para el desempeño de una empresa sino para su imagen corporativa, es por eso que la tolerancia a fallas es un tema que debe ser considerando muy seriamente por toda empresa que quiera ser o llegar a ser competitiva. También es importante mencionar que siempre se debe pensar en la escalabilidad que tendrá un proyecto relacionado con un clúster de alta disponibilidad, de tal manera que el desempeño del clúster no se vea degradado en el futuro por temas relacionados con la falta de recursos por la poca capacidad de los equipos.

Un clúster de Alta disponibilidad, así como cualquier otro equipo que esté conectado a una red de computadores, puede ser objeto de ataques por personas malintencionadas; por lo tanto deben implementarse las medidas de seguridad

necesarias para proteger a los componentes del clúster, como por ejemplo: control de acceso físico a los servidores, implementación de un firewall para el sistema operativo y los servicios del clúster, etc.

Para finalizar, quiero mencionar que nunca se debe confundir la palabra *libre* con *gratis*, ni tampoco se debe asumir que, por implementar un proyecto usando software libre el proyecto no va a tener ningún coste; en la práctica, sólo el software es el que se distribuye libre y gratuitamente, mas la instalación, configuración, mantenimiento y soporte del software libre en algunas casos, por no decir en su gran mayoría, sí tendrá un coste.

## BIBLIOGRAFÍA

El sistema de archivos en clúster conforma una parte importante en la estrategia propuesta pues permite usar un disco compartido para múltiples computadores.

Bautts T.; Dawson T.; Purdy G. (2005). "Linux Network Administrator's Guide". O'Reilly Media, Inc.

Bookman, Charles. (2002). "Linux Clustering: Building and Maintaining Linux Clusters". Sams, 1era edición.

Cluster Labs. (2011). "Welcome to Pacemaker". Obtenido en línea el 23 de Agosto del 2011. Disponible en:  
[http://www.clusterlabs.org/wiki/Main\\_Page](http://www.clusterlabs.org/wiki/Main_Page)

Corosync Group. "Welcome". (2011). Corosync Group, Honza. Obtenido en línea el 16 de Agosto del 2011. Disponible en:  
<http://www.corosync.org/doku.php?id=welcome>

Diccionario Oxford. (2002). "Diccionario de Internet". Editorial Complutense.

Evan, Marcus. (2003). "Blueprints for High Availability". Wiley.

Free Software Foundation. (2008). "What is GNU?". Obtenido en línea el 4 de Agosto del 2011. Disponible en:  
<http://www.gnu.org/home.es.html>

Hwang, K.; Xu, Z. (2008). "Scalable parallel computing". WCB/McGraw-Hill.

Kooper, Karl. (2005). "The Linux Enterprise Cluster". No Starch Press.

Krakowiak, Sacha. (2003). "What is Middleware". ObjectWeb. Obtenido en línea el 11 de Agosto del 2011. Disponible en: <http://middleware.objectweb.org/>

LINBIT HA-Solutions GmbH. (2011). "Part I. Introduction to DRBD". Obtenido en línea el 7 de Noviembre del 2011. Disponible en: <http://www.drbd.org/docs/introduction/>

Linux-HA. (2011) "High Availability Software for Linux and other Platforms since 1999". Obtenido en línea el 11 de Noviembre del 2011. Disponible en: [http://www.linux-ha.org/wiki/Main\\_Page](http://www.linux-ha.org/wiki/Main_Page)

Lucke, Robert W. (2004). "Building Clustered Linux Systems". Prentice Hall, 1era edición.

Matthew, N.; Stones, R. (2011). "Beginning Linux Programming". 4ta. Edición. John Wiley & Sons.

Microsoft Corporation. (2008). "Función Servidor DNS". TechNet. Obtenido en línea el 12 de Diciembre del 2011. Disponible en: <http://technet.microsoft.com/es-es/library>

Oggerino, Chris. (2001). "High Availability Network Fundamentals". Cisco Press.

Orozco, A. Martha. (2006). "Informática Uno". Cengage Learning Editores.

Pfister, F. Gregory. (1998). "In Search of Clusters". 2da Edición. Prentice Hall PTR.

Red Hat Inc. (2011). "Red Hat Enterprise Linux Instalation Guide". Red Hat Engineering Content Services.

Schlossnagle, Theo. (2006). "Scalable Internet Architectures". Sams.

Schmidt, Klaus. (2010). "High Availability and Disaster Recovery: Concepts, Design, Implementation". Springer.

Silberschatz, A.; Galvin, B. P.; Gagne, G. (2005). "Operating System Concepts". 7ma Edición. J. Wiley & Sons.

Spector, David. (2000). "Building Linux Clusters". O'Reilly Media.

The Linux Foundation. (2009). "What Is Linux: An Overview of the Linux Operating System". Obtenido en línea el 6 de Agosto del 2011. Disponible en:  
<https://www.linux.com/learn/new-user-guides/376-linux-is-everywhere-an-overview-of-the-linux-operating-system>

Vrenios, Alex. (2000). "Building Linux Clusters". O'Reilly.

## GLOSARIO DE TÉRMINOS

**Autenticación:** Es el proceso de confirmación de algo o alguien como auténtico.

**Checksum:** Es el valor utilizado para verificar que los datos no contienen errores.

**Dirección IP:** Es el número que identifica a un computador dentro de una red que usa el protocolo IP.

**Distribución Linux:** Es la distribución de software basada en el núcleo Linux.

**Encriptación:** Es un procedimiento por el cual un mensaje es alterado de tal manera que sea incompresible para quien no esté autorizado a recibirla.

**Failover:** Significa tolerancia a fallos y se usa para determinar la capacidad de un sistema para seguir funcionando aún cuando se produce un fallo.

**Firewall:** Significa cortafuegos y se usa para aquellos sistemas encargados de bloquear los accesos no autorizados en una red.

**FTP:** Es el acrónimo inglés de *File Transfer Protocol* cuya traducción al español es *Protocolo de Transferencia de Archivos* y es como su nombre lo indica es el protocolo que se usa en la transferencia de archivos en una red TCP.

**HA:** Es el acrónimo inglés de *High Availability* cuya traducción al español es *Alta Disponibilidad* y se usa para indicar el grado de continuidad operacional de un sistema en un período de tiempo dado.

**HTTP:** Es el acrónimo inglés de *Hyper Text Transfer Protocol* cuya traducción al español es *Protocolo de Transferencia de Hipertexto* y es usado para acceder las páginas Web.

**Implementación:** Es la realización o ejecución de un plan o un diseño.

**Intercomunicación:** Es la capacidad de enviar y recibir mensajes entre dos o más computadores en una red TCP/IP.

**Interfaz:** Es el medio de conexión entre computadores.

**Internet:** Es la red mundial de computadores interconectados que usan el protocolo TCP/IP.

**IP:** Es el acrónimo inglés de *Internet Protocol* cuya traducción al español es *Protocolo de Internet* y es el protocolo encargado de la comunicación entre el origen y el destino en una red TCP/IP.

**iSCSI:** Es el acrónimo inglés de *Internet Small Computers System Interface* que quiere decir *Interfaz de Sistema para Computadores Pequeños en Internet* y es una interfaz estándar para la transmisión de datos.

**Migración:** Consiste en mover uno o más recursos de un nodo del clúster a otro.

**N:** La letra N en mayúscula puede ser un número entero.

**Nodo más Apropiado:** Es el nodo que posee el valor numérico más alto luego de sumar los valores de adherencia de todos los recursos dentro del clúster.

**Nodo:** Es un computador miembro del clúster.

**Página Web:** Es el acrónimo inglés de *World Wide Web* cuya traducción al español es *Red Informática Mundial* y consiste en un documento basado en hipertexto que permite.

**POP:** Es el acrónimo inglés de *Post Office Protocol* que quiere decir *Protocolo de la Oficina de Correo* y es el protocolo que permite obtener los mensajes de correo recibidos por un servidor de correo.

**Protocolo:** Es un conjunto de reglas establecidas para la comunicación entre computadores.

**Proxy:** Es un programa o dispositivo encargado de realizar una tarea en representación de otro.

**Replicación:** Consiste en copiar o duplicar los datos de un nodo a otro.

**Servicio:** Es un conjunto de actividades que responden a las necesidades de un cliente.

**Servidor:** Es una computador que provee uno o más servicios o otras computadores.

**Sincronización:** Es el proceso en el cual dos dispositivos se actualizan de manera que los dos contengan los mismos datos.

**SMTP:** Es el acrónimo inglés de *Simple Mail Transfer Protocol* cuya traducción al español es *Protocolo Simple para la Transferencia de Correo* y es usado para enviar y recibir correos a través de Internet.

**STONITH:** Es el acrónimo inglés de *Shoot The Other Node In The Head* cuya traducción al español es *Disparar al Otro Nodo en la Cabeza* y es usado como mecanismo de protección cuando se produce un conflicto entre dos o más nodos dentro del clúster.

**TCP/IP:** Es conjunto de protocolos que hacen posible la comunicación entre computadores.

**TCP:** Son las siglas del término en inglés *Transmission Control Protocol* cuya traducción al español es *Protocolo de*

*Control de Transmisión* y es el protocolo encargado de garantizar que los mensajes sean entrados en su destino.

**Usabilidad:** Es la facilidad con la que una persona puede usar una herramienta en particular.

**Valor de Adherencia:** Es valor numérico que determina si un recurso debe o no ser migrado a otro nodo.

## ANEXO 1 INSTALACIÓN Y CONFIGURACIÓN DEL CENTOS

### 1. INSTALACIÓN DEL SISTEMA OPERATIVO

El siguiente procedimiento de instalación de CentOS aplica para todos los nodos del clúster.

En la primera pantalla que muestra el instalador de CentOS, seleccionamos la primera opción para iniciar la instalación.



Mientras se carga el sistema de archivos temporal aparece una ventana como esta:

```
TCP cubic registered
Initializing XFRM netlink socket
NET: Registered protocol family 17
registered taskstats version 1
IMA: No TPM chip found, activating TPM-bypass!
rtc_cmos 00:04: setting system clock to 2011-07-24 14:16:58 UTC (1311517018)
Initializing network drop monitor service
Freeing unused kernel memory: 1220k freed
Write protecting the kernel read-only data: 7176k
input: ImPS/2 Generic Wheel Mouse as /devices/platform/i8042/serio1/input/input3

Greetings.
anaconda installer init version 13.21.82 starting
mounting /proc filesystem... done
creating /dev filesystem... done
starting udev...done
mounting /dev/pts (unix98 pty) filesystem... done
mounting /sys filesystem... done
trying to remount root filesystem read write... done
mounting /tmp as tmpfs... done
running install...
running /sbin/loader
detecting hardware...
waiting for hardware to initialize...
-
```

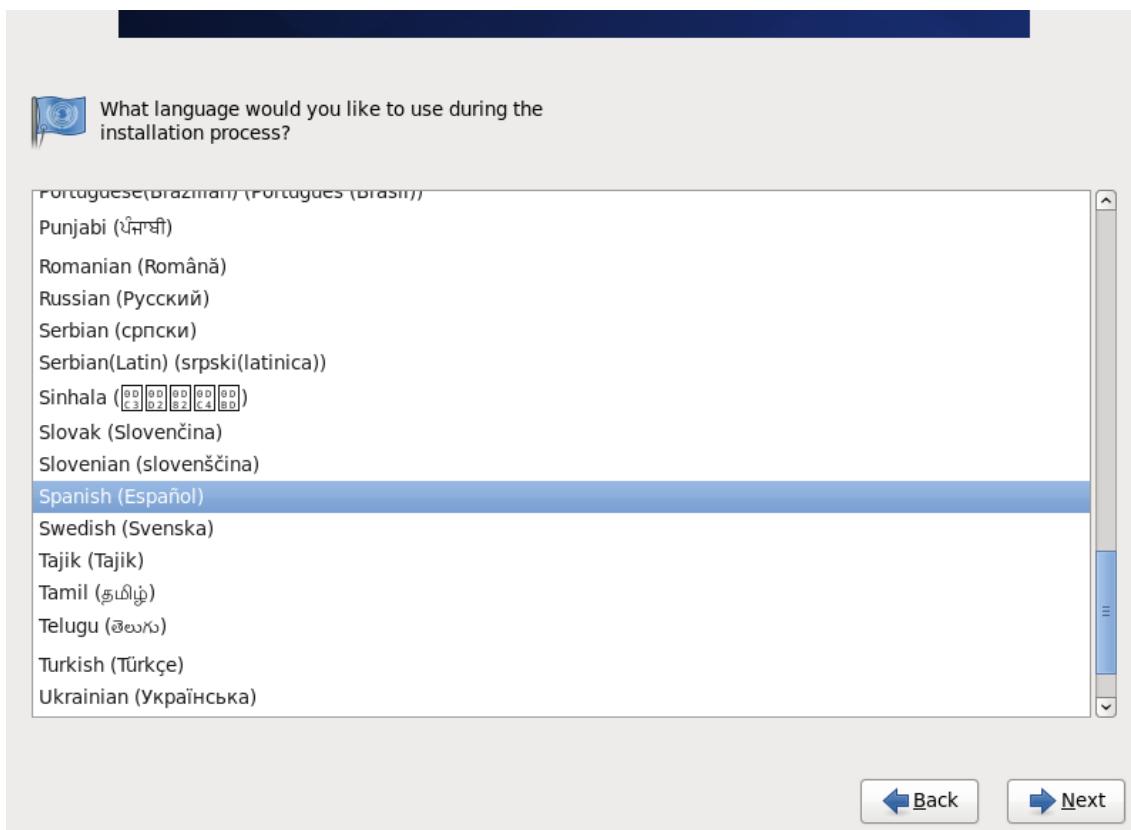
## 2. VERIFICACIÓN DE INTEGRIDAD DEL INSTALADOR

Seleccionar la opción **skip** para omitir la verificación del DVD de instalación.



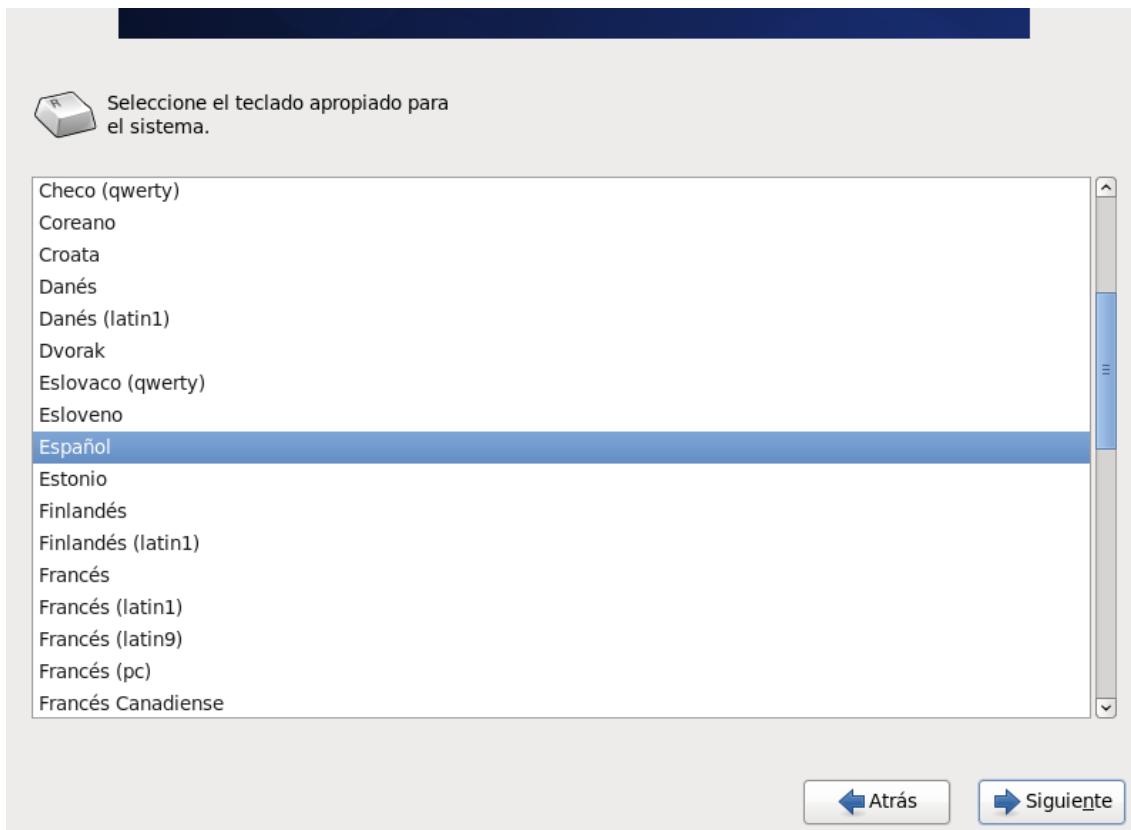
### 3. CONFIGURACIÓN DE LA INTERFAZ DE INSTALACIÓN

Seleccionar la opción español para cambiar el idioma de la interfaz de instalación.



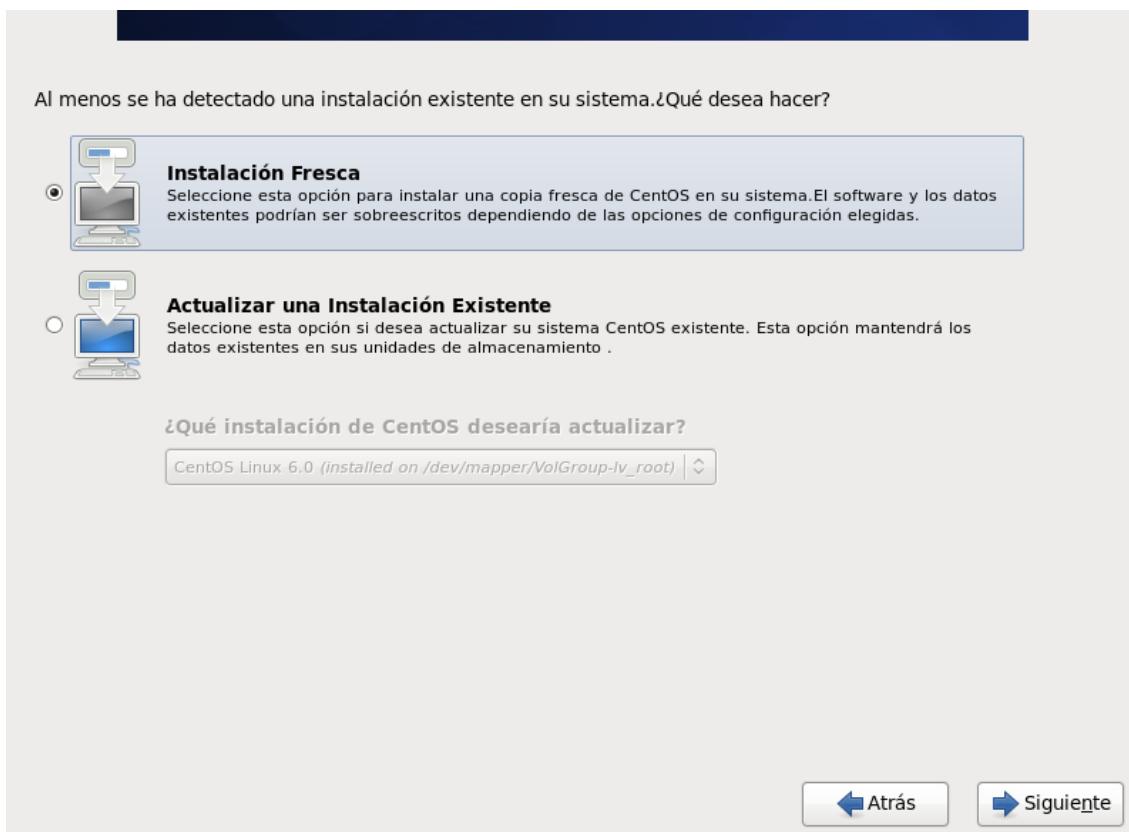
#### 4 . CONFIGURACIÓN DEL TECLADO

Seleccionar la opción español para cambiar el idioma del teclado.



## 5. SELECCIÓN INICIAL DE LA INSTALACIÓN

Seleccionar la opción **Instalación Fresca** para iniciar una instalación nueva del sistema operativo.



## 6. SELECCIÓN DEL TIPO DE ALMACENAMIENTO

Seleccionar la opción **Dispositivos de almacenamiento básicos**. En el caso de usar medios de almacenamiento avanzados es necesario seleccionar la segunda opción.

¿Qué tipo de dispositivos involucra su instalación?

**Dispositivos de almacenamiento básicos**

Instalaciones o actualizaciones para tipos comunes de dispositivos de almacenamiento. Si usted no está seguro de la opción apropiada para usted, ésta es probablemente la correcta.

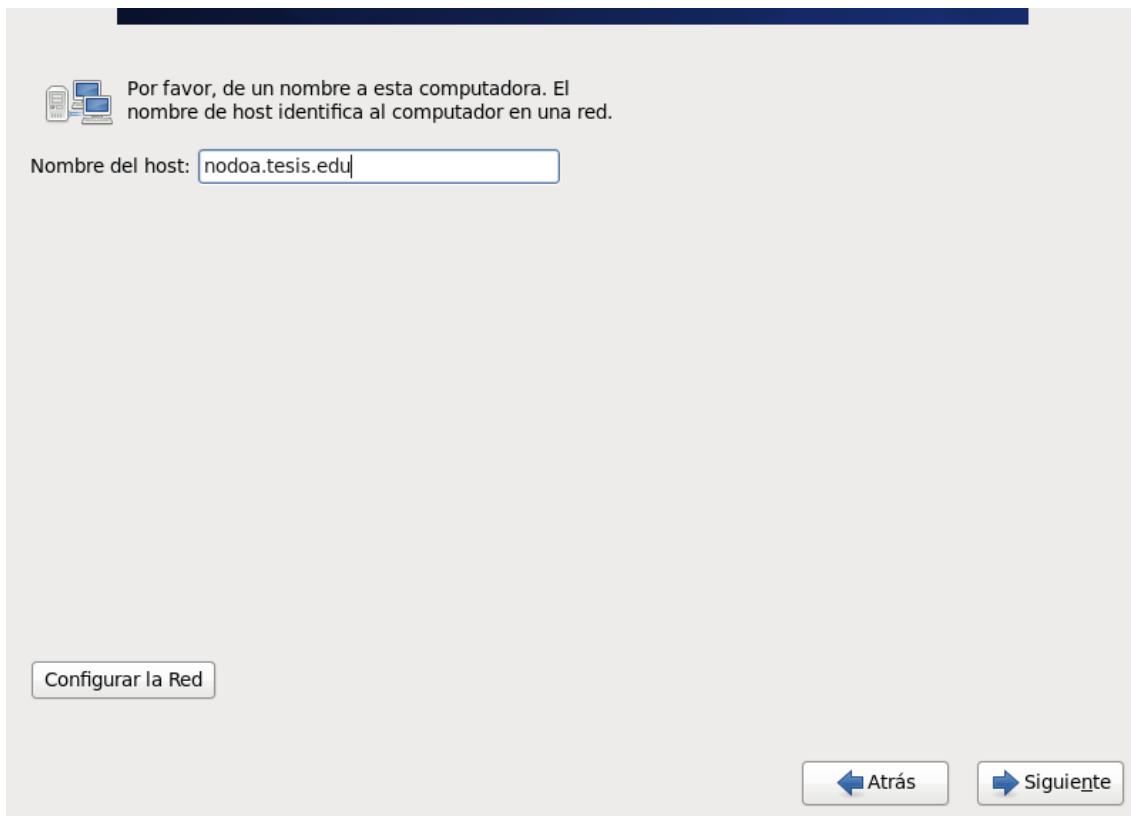
**Dispositivos de almacenamiento especializados**

Instala o actualiza dispositivos de empresa tales como Redes de área de almacenamiento (SAN). Esta opción le permitirá añadir discos FCoE / iSCSI / zFCP y filtrar los dispositivos que el instalador debe ignorar.

[!\[\]\(d3286db708ec09a3bd03986590d8861d\_img.jpg\)](#) [!\[\]\(0146fa9b09486178e272021adaa967d6\_img.jpg\)](#)

## 7 . NOMBRE DEL HOST

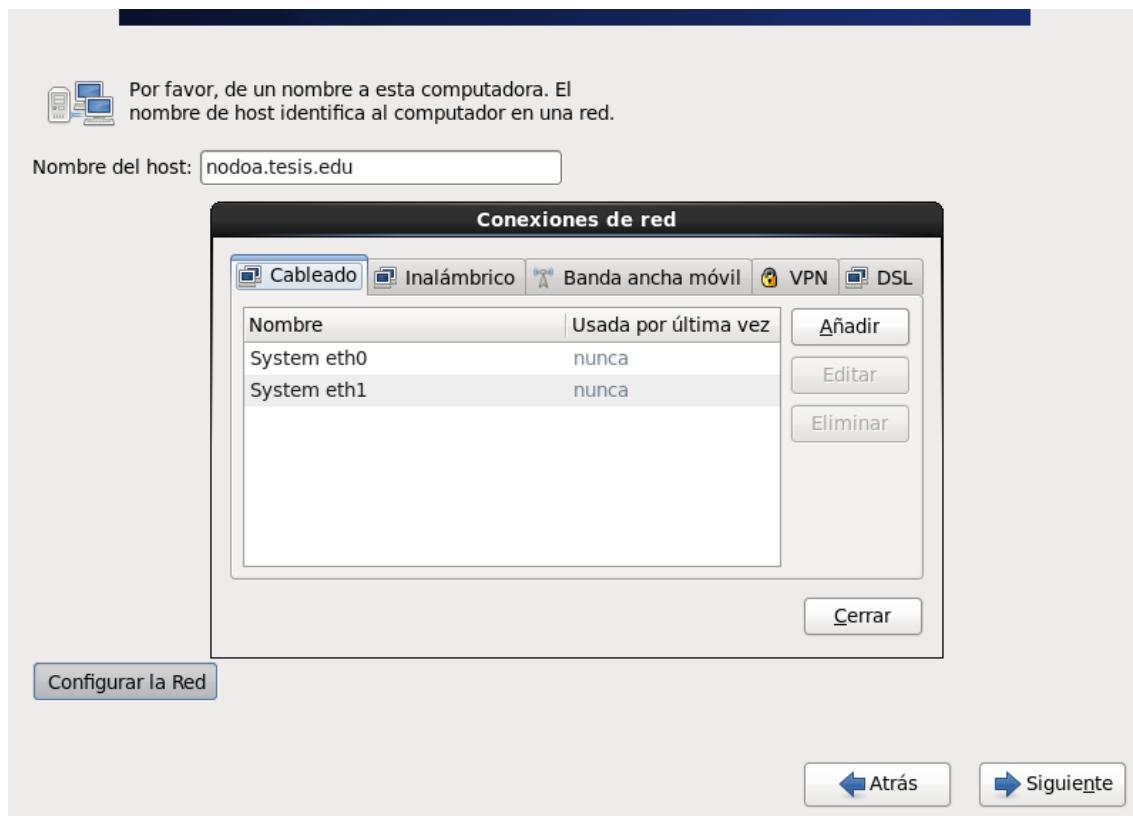
Seleccionar un nombre de host para el servidor. Para este ejemplo el nombre del host será: **nodoa.thesis.edu**



## 8 . CONFIGURACIÓN DE RED

Seleccionar la tarjeta de red respectiva y luego la opción **Editar** para configurar los parámetros de red para cada tarjeta de red.

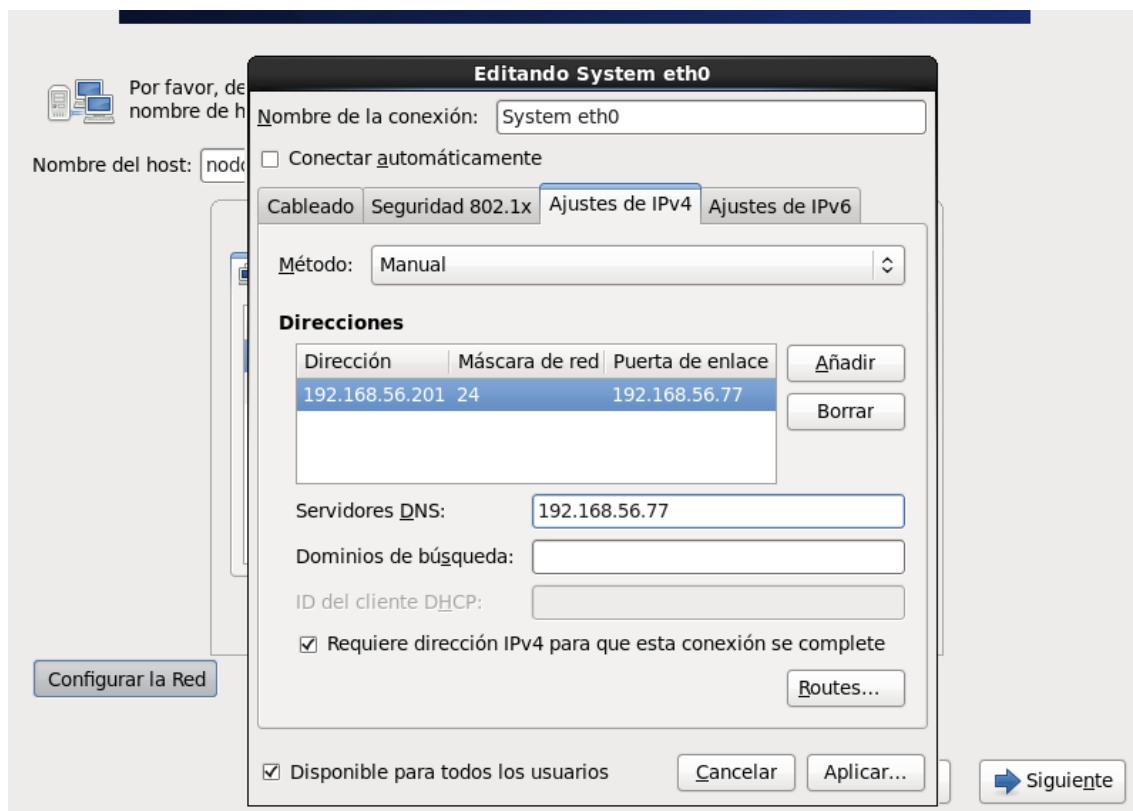
También es posible configurar otros parámetros de red para redes inalámbricas, Banda ancha, VPN ó DLS.



En la pestaña de **Ajustes de IPv4** seleccionar la opción **Añadir** y escribir la nueva dirección IP<sup>6</sup> del servidor, la máscara de red y la dirección IP del Gateway.

En el campo **Servidores DNS** escribir la dirección IP del servidor DNS principal.

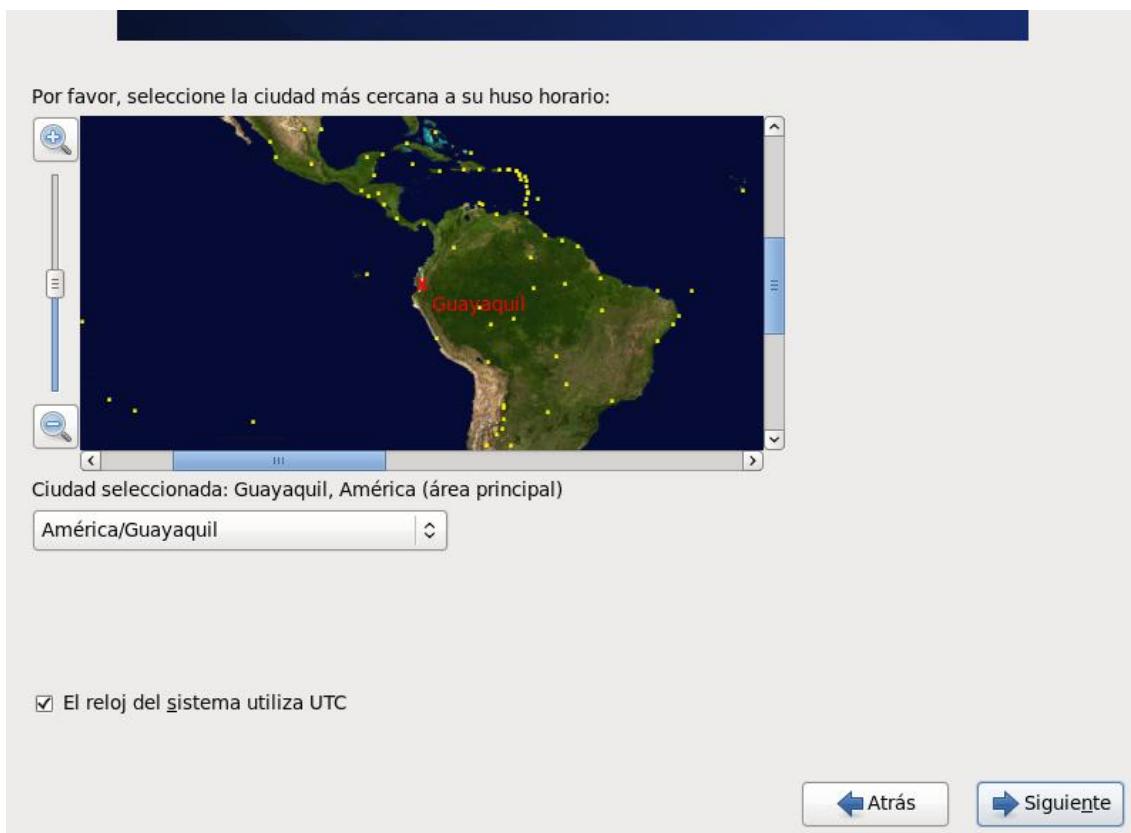
También es posible configurar los parámetros del cableado, seguridad e IPv6.



<sup>6</sup> Véase Glosario – Dirección IP

## 9. CONFIGURACIÓN DEL USO HORARIO Y LA HORA.

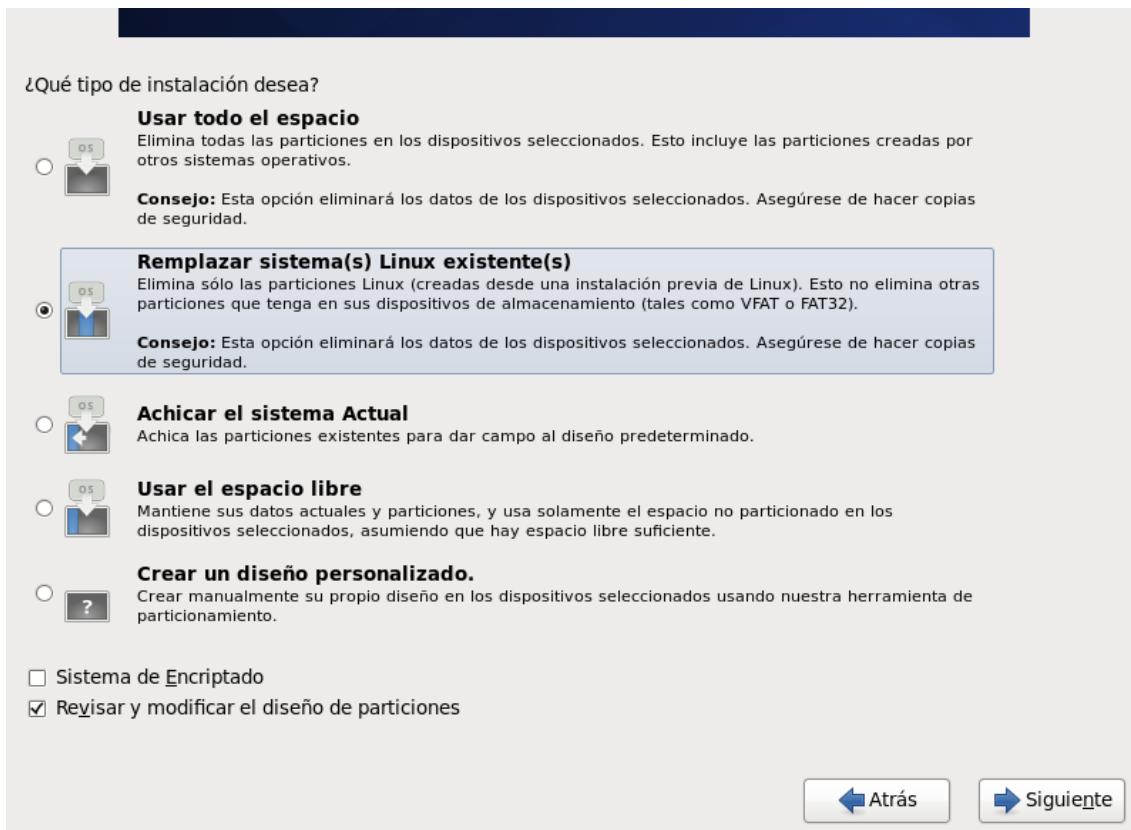
Seleccionar la opción **América/Guayaquil** para definir el uso horario y marcar la opción **El reloj del sistema utiliza UTC** para ajustar el reloj.



## 10. SELECCIÓN DEL TIPO DE INSTALACIÓN

Es posible seleccionar entre **Usar todo el espacio** ó **reemplazar sistemas(s) Linux existente** y así eliminar todas las particiones Linux existentes en el disco.

También es recomendable escoger la opción **Revisar y modificar el diseño de particiones** para obtener una descripción detallada de la estructura de las particiones antes de que sean creadas por el instalador.



## 11. CONFIGURACIÓN DE LAS PARTICIONES DEL DISCO

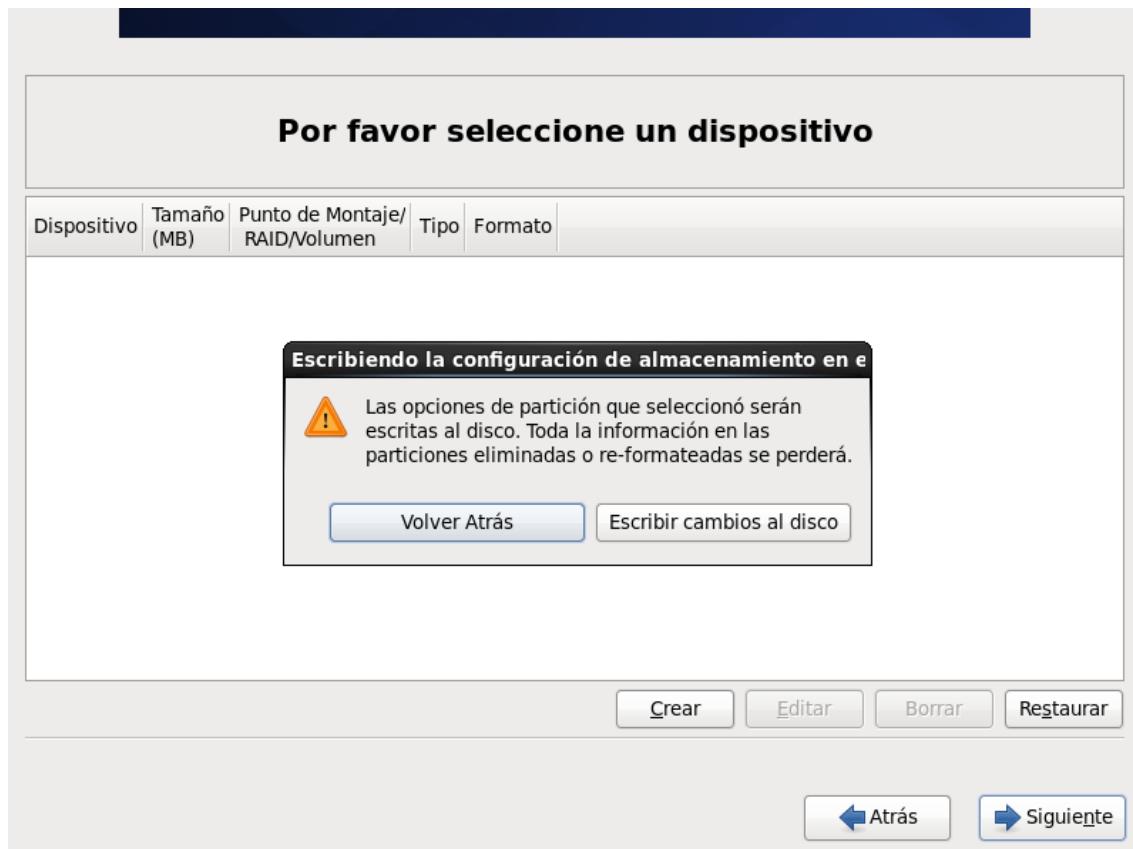
En esta pantalla es posible establecer las particiones del disco, sin embargo sólo para este ejemplo vamos a aceptar la configuración por default.

The screenshot shows a software interface for managing disk partitions. At the top, a bold black header reads "Please Select A Device". Below it is a table with columns: Device, Size (MB), Mount Point/RAID/Volume, Type, and Format. The table lists partitions from two drives: sda and sdb. Drive sda contains partitions sda1 through sda4. Drive sdb contains partitions sdb1 through sdb3. All partitions are of type ext4 and have a checkmark in the Format column, indicating they are formatted. The "Mount Point/RAID/Volume" column shows the mount points: /boot, /var, swap, / for sda1-4; and /disk1, /disk2, /disk3 for sdb1-3. The "Type" column shows ext4 for all partitions. The "Format" column has checkmarks for all entries. The "Device" column lists the device names followed by their respective paths (e.g., sda1, /dev/sda1). Below the table, there are four buttons: "Create", "Edit", "Delete", and "Reset".

Device	Size (MB)	Mount Point/RAID/Volume	Type	Format
<b>Hard Drives</b>				
▼ sda (/dev/sda)				
sda1	500	/boot	ext4	✓
sda2	5000	/var	ext4	✓
sda3	1024		swap	✓
sda4	13955	/	ext4	✓
▼ sdb (/dev/sdb)				
sdb1	3000	/disk1	ext4	✓
sdb2	3000	/disk2	ext4	✓
sdb3	4239	/disk3	ext4	✓

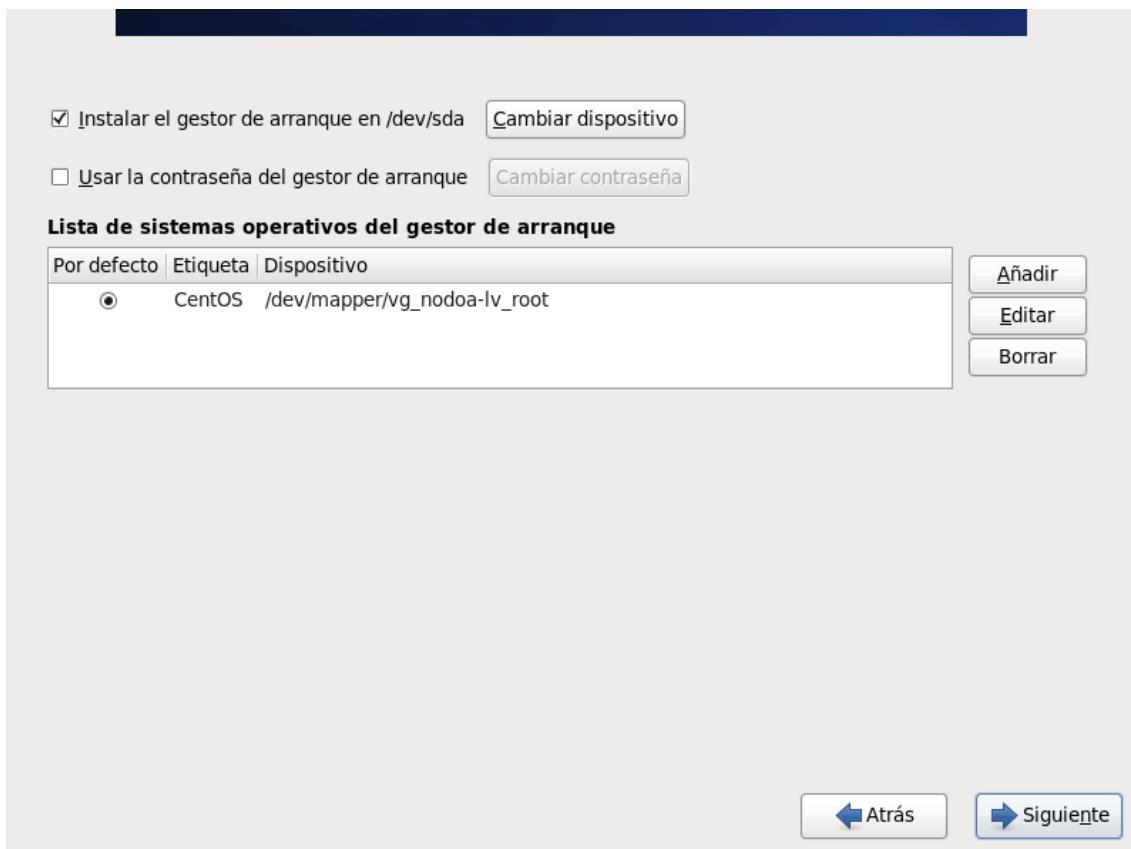
[Create](#) [Edit](#) [Delete](#) [Reset](#)

Seleccionar la opción **Escribir cambios al disco** para guardar la nueva estructura e iniciar el proceso de creación de las particiones.



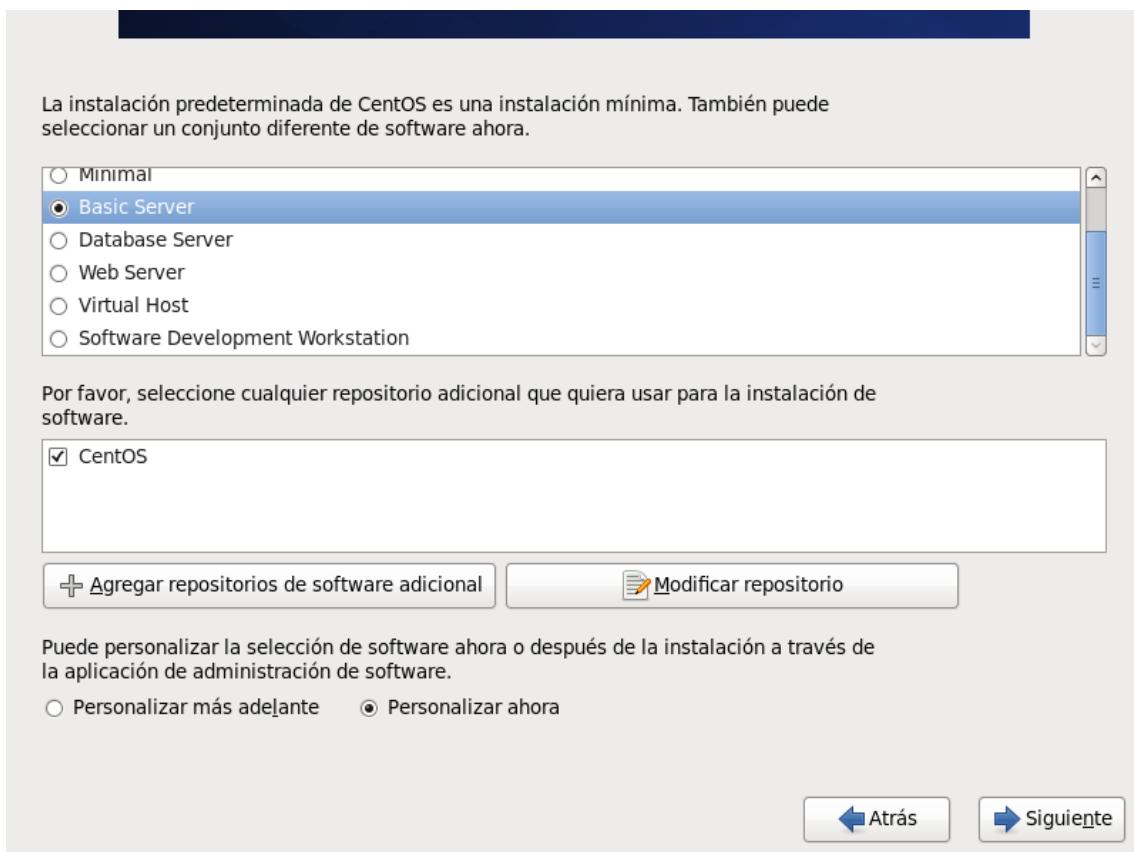
## 12. CONFIGURACIÓN DE ARRANQUE

Marcar la opción **Instalar el gestor de arranque en /dev/sda** y desmarcar la opción **Usar la contraseña del gestor de arranque.**



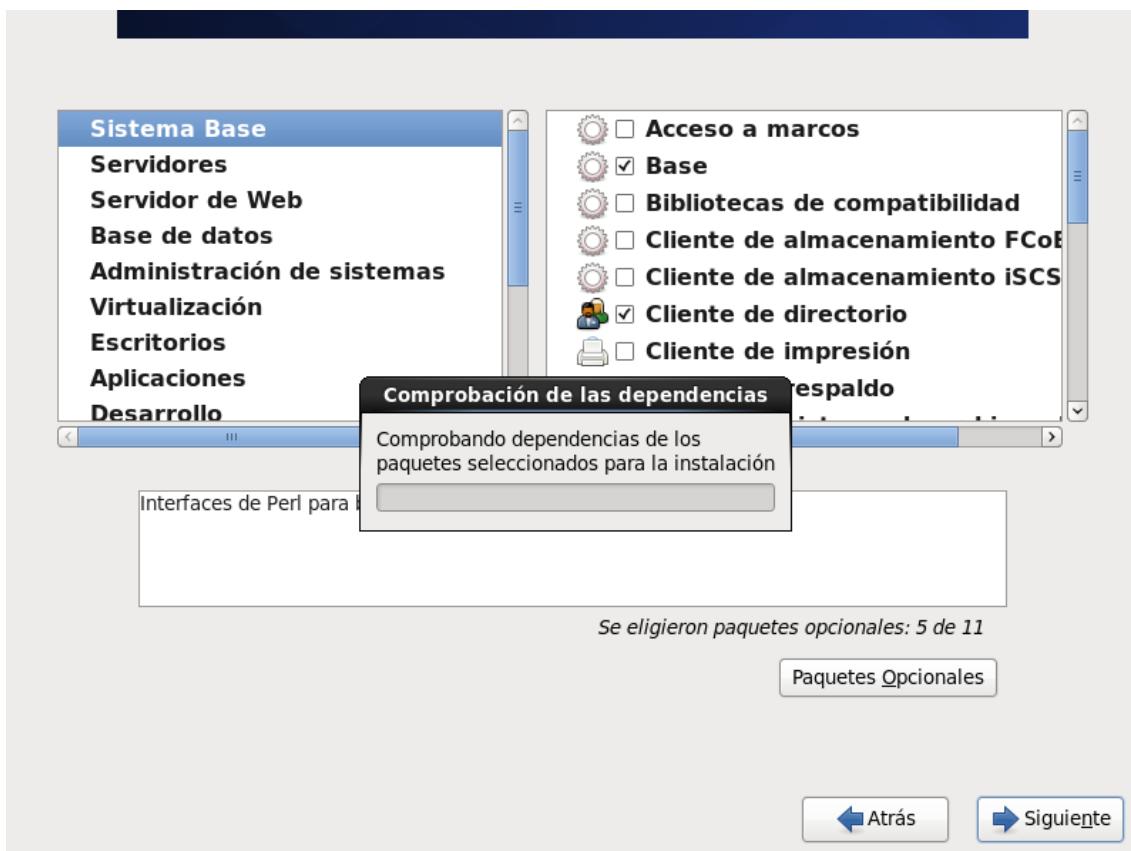
### 13. SELECCIÓN DEL TIPO DE SERVIDOR

Seleccionar la opción **Basic Server** para instalar sólo los paquetes necesarios del servidor sin sobrecargar el sistema de aplicaciones que no son indispensables.



## 14. SELECCIÓN DE PAQUETES INDIVIDUALES

En esta pantalla es posible agregar o quitar paquetes de la instalación por default.



## 15. FIN DE LA INSTALACIÓN

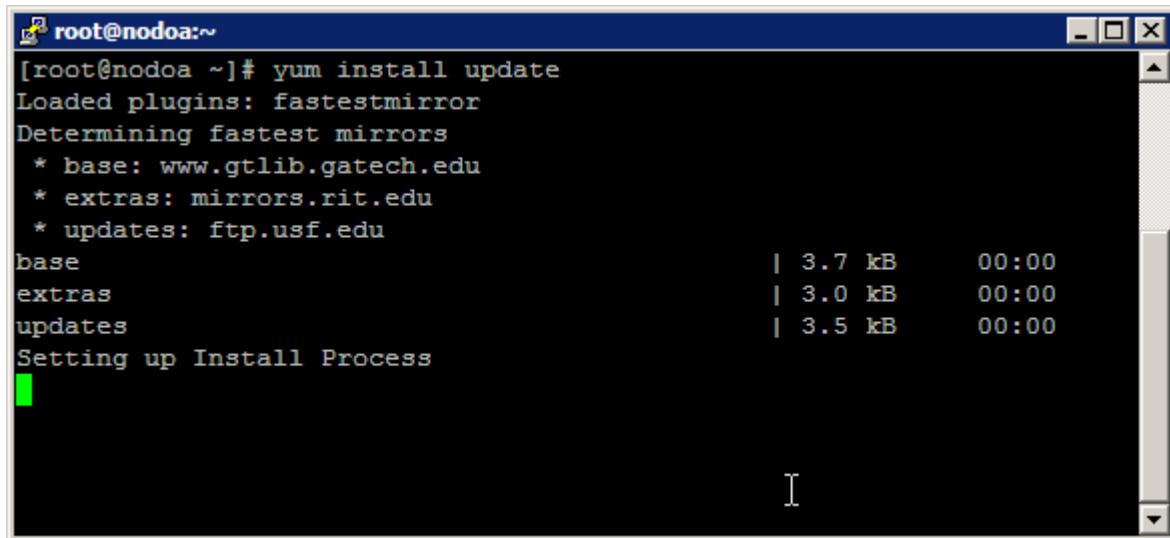
Es esta pantalla que indica la finalización del proceso de instalación. Ahora sólo hay que remover el disco de instalación y reiniciar el servidor.



Repetir el mismo procedimiento en el nodo B

## 16. ACTUALIZACIÓN DEL SISTEMA

Para instalar las actualizaciones disponibles en CentOS es necesario ejecutar el comando **yum install update**.



The screenshot shows a terminal window with a blue title bar containing the text "root@nodoa:~". The main area of the terminal displays the output of the command "yum install update". The output includes the following text:  
[root@nodoa ~]# yum install update  
Loaded plugins: fastestmirror  
Determining fastest mirrors  
\* base: www.gtlib.gatech.edu  
\* extras: mirrors.rit.edu  
\* updates: ftp.usf.edu  
base | 3.7 kB 00:00  
extras | 3.0 kB 00:00  
updates | 3.5 kB 00:00  
Setting up Install Process  
[progress bar]

También es posible verificar si existen actualizaciones disponibles para luego instalarlas individualmente.

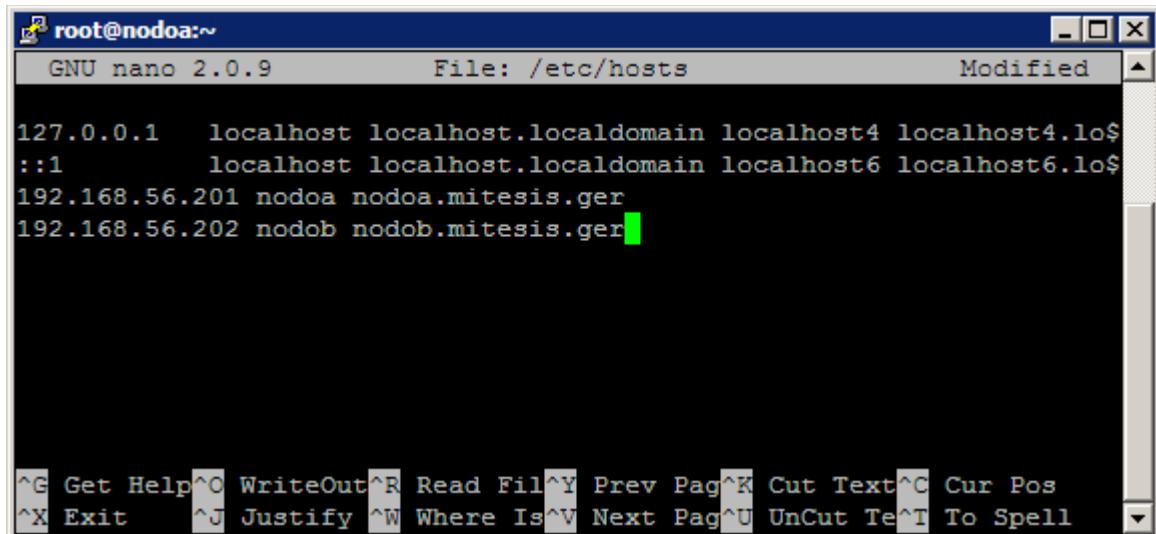
```
#yum check-update  
#yum install <nombre_del_paquete>
```

**Repetir el mismo procedimiento en el nodo B**

## 17. CONFIGURACIÓN DEL ARCHIVO /ETC/HOSTS

Es necesario agregar las siguientes líneas al archivo /etc/hosts.

```
192.168.56.201 nodoa nodoa.mitesis.ger
192.168.56.202 nodob nodob.mitesis.ger
```



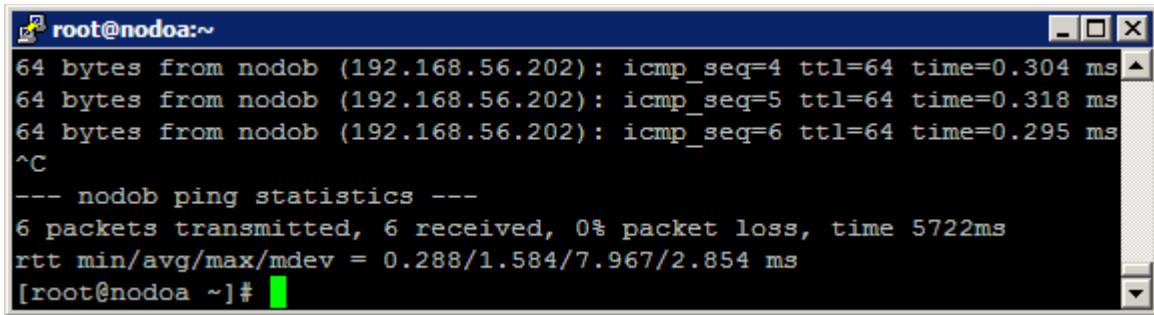
```
root@nodoa:~#
GNU nano 2.0.9           File: /etc/hosts          Modified
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.lo$  
::1         localhost localhost.localdomain localhost6 localhost6.lo$  
192.168.56.201 nodoa nodoa.mitesis.ger  
192.168.56.202 nodob nodob.mitesis.ger

^G Get Help ^O WriteOut ^R Read Fil^Y Prev Pag^K Cut Text^C Cur Pos
^X Exit      ^J Justify ^W Where Is^V Next Pag^U UnCut Te^T To Spell
```

## 18. PRUEBA DE CONECTIVIDAD DE RED

Para verificar que hemos editado correctamente el archivo /etc/hosts ejecutamos el siguiente comando:

```
ping nodoa
```



```

root@nodoa:~#
64 bytes from nodob (192.168.56.202): icmp_seq=4 ttl=64 time=0.304 ms
64 bytes from nodob (192.168.56.202): icmp_seq=5 ttl=64 time=0.318 ms
64 bytes from nodob (192.168.56.202): icmp_seq=6 ttl=64 time=0.295 ms
^C
--- nodob ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5722ms
rtt min/avg/max/mdev = 0.288/1.584/7.967/2.854 ms
[root@nodoa ~]#

```

**Repetir el mismo procedimiento en el nodo B**

## 19. CONFIGURACIÓN DEL REPOSITORIO DE RED HAT

Para agregar el repositorio de Red Hat es necesario ejecutar los siguientes comandos:

```
# su -c 'rpm -Uvh http://dl.fedoraproject.org/pub/epel/5/i386/epel-release-5-3.noarch.rpm'
```

```
#wget -O /etc/yum.repos.d/pacemaker.repo
http://clusterlabs.org/rpm/epel-5/clusterlabs.repo
```

**Repetir el mismo procedimiento en el nodo B**

## ANEXO 2 INSTALACIÓN Y CONFIGURACIÓN DE DRBD

### 1. INSTALACIÓN DE DRBD

Para instalar DRBD es necesario ejecutar el siguiente comando:

```
yum install -y drbd83 kmod-drbd83
```

**Repetir el mismo procedimiento en el nodo B**

### 2. CONFIGURACIÓN BÁSICA DE DRBD

Abrir y editar el archivo /etc/drbd.conf manualmente.

```
include "drbd.d/basico.conf";
include "drbd.d/*.*";
```

### 3. CREAR EL ARCHIVO DE CONFIGURACIÓN DE RECURSOS

Crear el archivo /etc/drbd.d/r0.res con las siguientes líneas.

```

resource r0 {
    volume 0 {
        device /dev/drbd0;
        disk /dev/sdb1;
        meta-disk internal;
    }
    on nodoa.mitesis.ger {
        address 192.168.56.201:7789;
    }
    on nodob.mitesis.ger {
        address 192.168.56.202:7789;
    }
}

```

#### **4. COPIAR LOS ARCHIVOS DE CONFIGURACIÓN AL NODO B**

Copiar la configuración de drbd del nodo A al nodo B

```

scp /etc/drbd.d/* nodob:/etc/drbd.d
scp /etc/drbd.conf nodob:/etc/

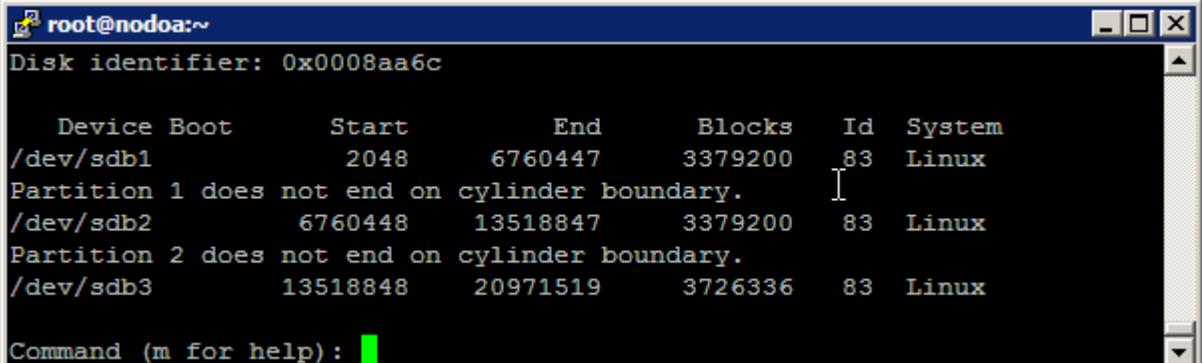
```

#### **5. CREAR PARTICIONES CON FDISK**

Crear 3 particiones primarias en la unidad /dev/sdb con fdisk.

```
#fdisk /dev/sdb
```

Una vez creadas las particiones tendremos el disco configurado de la siguiente manera:



```

root@nodoa:~#
Disk identifier: 0x0008aa6c

      Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048       6760447     3379200   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sdb2       6760448      13518847     3379200   83  Linux
Partition 2 does not end on cylinder boundary.
/dev/sdb3      13518848      20971519     3726336   83  Linux

Command (m for help): 

```

Repetir el mismo procedimiento en el nodo B

## 6. INICIALIZAR LAS PARTICIONES

Una vez creadas las particiones hay que ejecutar las siguientes instrucciones

```

dd if=/dev/zero of=/dev/sdb1 bs=1M count=128
dd if=/dev/zero of=/dev/sdb2 bs=1M count=128
dd if=/dev/zero of=/dev/sdb3 bs=1M count=128

```

Repetir el mismo procedimiento en el nodo B

## 7. CREAR LOS METADATOS PARA LAS UNIDADES DRBD

Ejecutar el siguiente comando para crear los metadatos en todas las unidades drbd

```
drbdadm create-md all
```

**Repetir el mismo procedimiento en el nodo B**

## 8. INICIAR EL SERVICIO

Para arrancar el servicio DRBD es necesario ejecutar el siguiente comando:

```
service drbd start
```

**Repetir el mismo procedimiento en el nodo B**

## 9. ESTABLECER LAS UNIDADES COMO PRIMARIAS

Para establecer las unidades del nodo A como unidades primarias hay que ejecutar el siguiente comando:

```
drbdadm --overwrite-data-of-peer primary all
```

Este paso debe ser ejecutado sólo en el nodo A

## 10. FORMATEAR LAS PARTICIONES CON EL SISTEMA EXT4

Ejecutar las siguientes líneas para formatear las particiones:

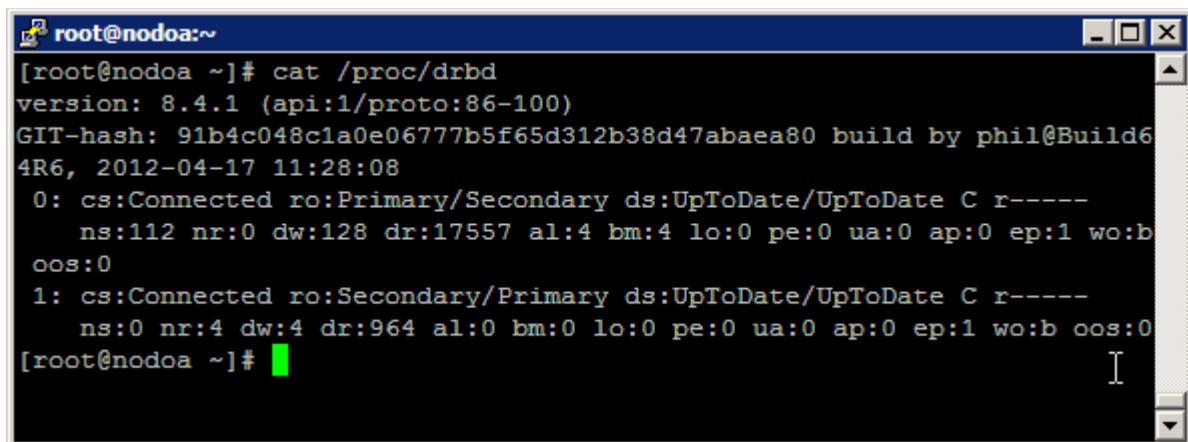
```
mkfs.ext4 /dev/drbd0  
mkfs.ext4 /dev/drbd1  
mkfs.ext4 /dev/drbd2
```

Este paso debe ser ejecutado sólo en el nodo A

## 11. PRUEBAS DE DRBD

Para verificar que DRBD se ha iniciado correctamente, ejecutamos los siguientes comandos:

```
cat /proc/drbd
```



```
root@nodoa:~# cat /proc/drbd
version: 8.4.1 (api:1/proto:86-100)
GIT-hash: 91b4c048c1a0e06777b5f65d312b38d47abaea80 build by phil@Build6
4R6, 2012-04-17 11:28:08
  0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
    ns:112 nr:0 dw:128 dr:17557 al:4 bm:4 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b
    oos:0
  1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
    ns:0 nr:4 dw:4 dr:964 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0
[root@nodoa ~]#
```

## ANEXO 3 INSTALACIÓN Y CONFIGURACIÓN DE COROSYNC

### 1. INSTALACIÓN DE COROSYNC

Para instalar Corosync es necesario ejecutar el siguiente comando:

```
yum -y install corosync
```

**Repetir el mismo procedimiento en el nodo B**

### 2. GENERACIÓN DE LA CLAVE DE COROSYNC

Para generar la clave de Corosync es necesario ejecutar el comando:

```
corosync-keygen
```

### 3. CONFIGURACIÓN BÁSICA DE COROSYNC

Crear el archivo /etc/corosync/corosync.conf con las siguientes líneas.

```

compatibility: whitetank

totem {
    version: 2
    secauth: off
    threads: 0
    interface {
        ringnumber: 0
        bindnetaddr: 192.168.56.0
        mcastaddr: 226.94.1.1
        mcastport: 7777
    }
}

logging {
    fileline: off
    to_stderr: yes
    to_logfile: yes
    to_syslog: yes
    logfile: /var/log/corosync.log
    debug: off
    timestamp: on
    logger_subsys {
        subsys: AMF
        debug: off
    }
}

amf {
    mode: disabled
}

```

#### **4. COPIAR LOS ARCHIVOS DE CONFIGURACIÓN AL NODO B**

Para que los nodos puedan conectarse al clúster, estos deben compartir los mismos archivos de configuración. Para ellos hay que ejecutar los siguientes comandos:

```
scp /etc/corosync/* nodob:/etc/corosync/  
scp /etc/corosync/authkey nodob:/etc/corosync/
```

## 5. INICIAR EL SERVICIO COROSYNC

Para arrancar el servicio Corosync es necesario ejecutar el siguiente comando:

```
service corosync start
```

**Repetir el mismo procedimiento en el nodo B**

## 6. PRUEBAS DE COROSYNC

Para verificar que Corosync se ha iniciado correctamente, ejecutamos los siguientes comandos:

```
grep -e "Corosync Cluster Engine" -e "configuration file"  
/var/log/messages
```

```
root@nodoa:/etc/drbd.d
[root@nodoa drbd.d]# grep -e "Corosync Cluster Engine" -e "configuration
file" /var/log/messages
Apr 29 10:27:33 nodoa corosync[4032]: [MAIN ] Corosync Cluster Engine
('1.4.1'): started and ready to provide service.
Apr 29 10:27:33 nodoa corosync[4032]: [MAIN ] Successfully read main
configuration file '/etc/corosync/corosync.conf'.
[root@nodoa drbd.d]#
```

```
grep TOTEM /var/log/messages
```

```
root@nodoa:/etc/drbd.d
[root@nodoa drbd.d]# grep TOTEM /var/log/messages
Apr 29 10:27:33 nodoa corosync[4032]: [TOTEM ] Initializing transport
(UDP/IP Multicast).
Apr 29 10:27:33 nodoa corosync[4032]: [TOTEM ] Initializing transmit/r
eceive security: libtomcrypt SOBER128/SHA1HMAC (mode 0).
Apr 29 10:27:33 nodoa corosync[4032]: [TOTEM ] The network interface [
192.168.56.201] is now up.
Apr 29 10:27:34 nodoa corosync[4032]: [TOTEM ] Process pause detected
for 732 ms, flushing membership messages.
Apr 29 10:27:34 nodoa corosync[4032]: [TOTEM ] A processor joined or l
eft the membership and a new membership was formed.
[root@nodoa drbd.d]#
```

```
grep pcmk_startup /var/log/messages
```

```
root@nodoa:/etc/drbd.d [root@nodoa drbd.d]# grep pcmk_startup /var/log/messages
Apr 29 10:27:34 nodoa corosync[4032]: [pcmk ] info: pcmk_startup: CRM
: Initialized
Apr 29 10:27:34 nodoa corosync[4032]: [pcmk ] Logging: Initialized pc
mk_startup
Apr 29 10:27:34 nodoa corosync[4032]: [pcmk ] info: pcmk_startup: Max
imum core file size is: 18446744073709551615
Apr 29 10:27:34 nodoa corosync[4032]: [pcmk ] info: pcmk_startup: Ser
vice: 10
Apr 29 10:27:34 nodoa corosync[4032]: [pcmk ] info: pcmk_startup: Loc
al hostname: nodoa.mitesis.ger
[root@nodoa drbd.d]#
```

## ANEXO 4 INSTALACIÓN Y CONFIGURACIÓN DE PACEMAKER

### 7. INSTALACIÓN DE PACEMAKER

Para instalar Pacemaker es necesario ejecutar el siguiente comando:

```
yum install -y pacemaker
```

Repetir el mismo procedimiento en el nodo B

### 8. CONFIGURACIÓN DE PACEMAKER COMO ADMINISTRADOR DE RECURSOS DE COROSYNC

Para indicarle a Corosync que Pacemaker será el administrador de recursos es necesario agregar las siguientes líneas al archivo de configuración de Corosync:

```
service {
    # Load the Pacemaker Cluster Resource Manager
    name: pacemaker
    ver: 0
}
```

Repetir el mismo procedimiento en el nodo B

## ANEXO 5 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR DNS

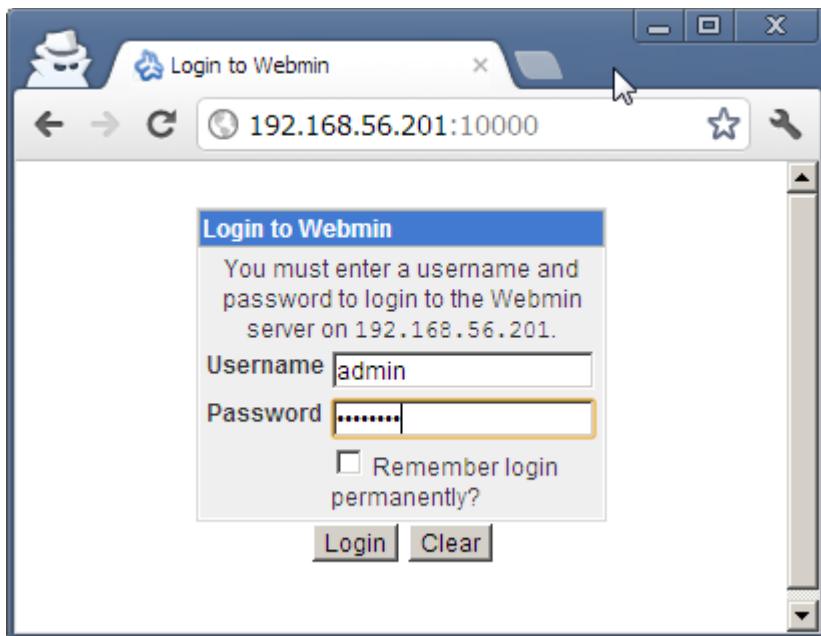
### 1. INSTALACIÓN DE WEBMIN

Para facilitar la configuración de los servicios como el servidor DNS o el servidor HTTP vamos a instalar Webmin usando el siguiente comando:

```
yum -y install webmin
```

Para acceder a Webmin usamos un navegador Web con los siguientes datos:

```
http://192.168.56.201:10000  
User: admin  
Password: admin
```



## 2. INSTALACIÓN DEL SERVIDOR DE DNS

Para instalar BIND como servidor DNS es necesario ejecutar el siguiente comando:

```
yum -y install bind-chroot
```

Repetir el mismo procedimiento en el nodo B

## 3. CREACIÓN DE LA ZONA MAESTRA

Para crear la zona maestra seleccionamos la opción Server y luego la opción BIND DNS Server.



Seleccionamos la opción Create Master Zone, completamos el formulario con la siguiente información.

Tipo de Zona  
Nombre de Dominio  
Nombre de Servidor  
Dirección de correo para notificaciones

Create Master Zone

New master zone options

Zone type	<input checked="" type="radio"/> Forward (Names to Addresses) <input type="radio"/> Reverse (Addresses to Names)
Domain name / Network	mitesis.ger
Records file	<input checked="" type="radio"/> Automatic <input type="radio"/> [ ] ...
Master server	nodoa.mitesis.ger <input checked="" type="checkbox"/> Add NS record for master server?
Email address	root@mitesis.ger
Use zone template?	<input type="radio"/> Yes <input checked="" type="radio"/> No IP address for template records
Add reverses for template addresses?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Refresh time	10800 seconds
Expiry time	604800 seconds
	Transfer retry time 3600 seconds
	Negative cache time 38400 seconds

#### 4. CREACIÓN DE LA ZONA INVERSA

Para crear la zona inversa seleccionamos la opción Create Master Zone, completamos el formulario con la siguiente información.

Tipo de Zona  
Nombre de Dominio  
Nombre de Servidor  
Dirección de correo para notificaciones

**Create Master Zone**

New master zone options

Zone type	<input type="radio"/> Forward (Names to Addresses) <input checked="" type="radio"/> Reverse (Addresses to Names)
Domain name / Network	192.168.56
Records file	<input checked="" type="radio"/> Automatic <input type="radio"/> [ ] ...
Master server	nodoa.mitesis.ger <input checked="" type="checkbox"/> Add NS record for master server?
Email address	root@mitesis.ger
Use zone template?	<input type="radio"/> Yes <input checked="" type="radio"/> No IP address for template records
Add reverses for template addresses?	<input checked="" type="radio"/> Yes <input type="radio"/> No
Refresh time	10800 seconds
Transfer retry time	3600 seconds
Expiry time	604800 seconds
Negative cache time	38400 seconds

**Create**

## 5. CREACIÓN DE LOS REGISTROS CON LAS DIRECCIONES

Para crear las direcciones es necesario seleccionar la opción Address y llenamos el formulario con la siguiente información.

The screenshot shows a software interface titled "Address Records" with a sub-header "In mitesis.ger". A dialog box is open for "Add Address Record". The fields are as follows:

- Name:** nodoa
- Address:** 192.168.56.201
- Time-To-Live:** Default (radio button selected), seconds dropdown set to 0
- Update reverse?**:
  - Yes (radio button selected)
  - Yes (and replace existing) (radio button)
  - No (radio button)

A "Create" button is at the bottom left of the dialog.

Hay que repetir el mismo procedimiento para en el nodob, dns, web y mail como se muestra en la figura.

**Address Records**

In mitesis.ger

**Add Address Record**

Name	<input type="text"/>	Time-To-Live	<input checked="" type="radio"/> Default	<input type="radio"/>	<input type="text"/>
Address	<input type="text"/>	...			
Update reverse?	<input checked="" type="radio"/> Yes	<input type="radio"/> Yes (and replace existing)	<input type="radio"/> No		

**Create**

Select all | Invert selection.

	Name	TTL	Address
<input type="checkbox"/>	nodoa.mitesis.ger.	Default	192.168.56.201
<input type="checkbox"/>	nodob.mitesis.ger.	Default	192.168.56.202
<input type="checkbox"/>	dns.mitesis.ger.	Default	192.168.56.250

	Name	TTL	Address
<input type="checkbox"/>	ftp.mitesis.ger.	Default	192.168.56.251
<input type="checkbox"/>	web.mitesis.ger.	Default	192.168.56.252

Select all | Invert selection.

**Delete Selected**  Delete reverses too?

## ANEXO 6 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR FTP

### 1. INSTALACIÓN DEL FTP

Para instalar vsFTPD como servidor de FTP es necesario ejecutar el siguiente comando:

```
yum -y install vsftpd
```

Repetir el mismo procedimiento en el nodo B

### 2. CONFIGURACIÓN DEL VSFTPD

Antes de arrancar el servicio de FTP hay editar el archivo /etc/vsftpd/vsftpd.conf y agregar la siguiente línea:

```
anon_root=/var/ftp/pub
```

## ANEXO 7 CONFIGURACIÓN DE LOS RECURSOS

### 1. CONFIGURACIÓN DE LOS RECURSOS EN PACEMAKER

Para registrar los recursos dentro del clúster vamos a crear un archivo temporal /root/config.txt donde agregaremos las siguientes líneas.

```

node nodoa.mitesis.ger
node nodob.mitesis.ger
primitive DNS_DAEMON lsb:named \
    operations $id="DNS_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \
    op monitor interval="15" timeout="15" start-delay="15" \
    meta target-role="started" is-managed="true"
primitive DISK1_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" directory="/disk1/" fstype="ext4" \
    operations $id="DISK1-operations" \
    op start interval="0" timeout="240" \
    op stop interval="0" timeout="100" \
    meta target-role="started" is-managed="true"
primitive DISK1_DRBD ocf:linbit:drbd \
    params drbd_resource="r0" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DNS_IP_ADDR ocf:heartbeat:IPAddr2 \
    params ip="192.168.56.250" cidr_netmask="24" \
    operations $id="DNS_IP_ADDR-operations" \
    op monitor start-delay="0" interval="30" \
    meta target-role="started" is-managed="true"
primitive FTP_DAEMON lsb:vsftpd \
    operations $id="FTP_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \
    op monitor interval="15" timeout="15" start-delay="15" \
    meta target-role="started" is-managed="true"
primitive FTP_IP_ADDR ocf:heartbeat:IPAddr2 \

```

```

params ip="192.168.56.251" cidr_netmask="24" \
operations $id="FTP_IP_ADDR-operations" \
op monitor start-delay="0" interval="30" \
meta target-role="started" is-managed="true"

group DISK1_DRBD_GROUP DISK1_DRBD
group DNS_AND_FTP_SERVICES DISK1_FOLDER DNS_IP_ADDR DNS_DAEMON
FTP_IP_ADDR FTP_DAEMON
ms MS_DISK1_DRBD_GROUP DISK1_DRBD_GROUP \
    meta master-max="1" master-node-max="1" clone-max="2" clone-
node-max="1" notify="true"
node-max="1" notify="true"
location LOC_DISK1_FOLDER DISK1_FOLDER -inf: nodoc.mitesis.ger
location LOC_DNS_IP_ADDR DNS_IP_ADDR -inf: nodoc.mitesis.ger
location LOC_FTP_IP_ADDR FTP_IP_ADDR -inf: nodoc.mitesis.ger
location LOC_MS_DISK1_DRBD_GROUP MS_DISK1_DRBD_GROUP -inf:
nodoc.mitesis.ger
colocation COL_DISK1_DRBD_GROUP inf: DNS_AND_FTP_SERVICES
MS_DISK1_DRBD_GROUP:Master
order RUN_DNS_AND_FTP_AFTER_DISK1_DRBD_GROUP inf:
MS_DISK1_DRBD_GROUP:promote DNS_AND_FTP_SERVICES:start
property $id="cib-bootstrap-options" \
    expected-quorum-votes="3" \
    stonith-enabled="false" \
    dc-version="1.1.6-3.el6-
a02c0f19a00cleb2527ad38f146ebc0834814558" \
    no-quorum-policy="ignore" \
    cluster-infrastructure="openais" \
    last-lrm-refresh="1336327391"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"

```

Ahora hay que ejecutar el siguiente comando para cargar el archivo de configuración en Pacemaker.

```
crm configure load replace config.txt
```

```
root@nodoa:~ [root@nodoa ~]# crm configure load replace config.txt
```

```
primitive DISK1_DRBD ocf:linbit:drbd \
    params drbd_resource="r0" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DISK1_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" directory="/disk1/" fstype="ext4"
\
operations $id="DISK1-operations" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="100" \
meta target-role="started" is-managed="true"
primitive DISK2_DRBD ocf:linbit:drbd \
    params drbd_resource="r1" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DISK2_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd1" directory="/disk2/" fstype="ext4"
\
operations $id="HTTP_FOLDER-operations" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="100" \
meta target-role="started" is-managed="true"
primitive DNS_DAEMON lsb:named \
    operations $id="DNS_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \
    op monitor interval="15" timeout="15" start-delay="15" \
    meta target-role="started" is-managed="true"
primitive DNS_IP_ADDR ocf:heartbeat:IPaddr2 \
    params ip="192.168.56.250" cidr_netmask="24" \
:
```

## **ANEXO 8 CONFIGURACIÓN DEL TERCER NODO**

### **1. INSTALACIÓN Y CONFIGURACIÓN DEL TERCER NODO DEL CLÚSTER**

Para agregar el tercer nodo al clúster es necesario repetir los pasos del Anexo 1 e instalar CentOS usando la siguiente información:

```
Nombre de host: nodoc.mitesis.ger  
Dirección IP: 192.168.56.203  
Máscara de Red: 255.255.255.0
```

También es necesario actualizar el archivo /etc/hosts en los tres nodos del clúster:

```
192.168.56.201 nodoa nodoa.mitesis.ger  
192.168.56.202 nodob nodob.mitesis.ger  
192.168.56.203 nodoc nodoc.mitesis.ger
```

### **2. CONFIGURACIÓN DE UN NUEVO RECURSO DE DRBD**

Para instalar DRBD hay que repetir los pasos del Anexo 2, pero esta vez es necesario crear el archivo /etc/drbd.d/r1.res en el nodoc para luego copiarlo al nodoa.

```
resource r1 {
    volume 0 {
        device /dev/drbd1;
        disk /dev/sdb2;
        meta-disk internal;
    }
    on nodoa.mitesis.ger {
        address 192.168.56.201:7790;
    }
    on nodoc.mitesis.ger {
        address 192.168.56.203:7790;
    }
}
```

### **3. INSTALACIÓN Y CONFIGURACIÓN DE COROSYNC**

Para instalar y configurar Corosync es necesario repetir todos los pasos del Anexo 3.

### **4. INSTALACIÓN Y CONFIGURACIÓN DE PACEMAKER**

Para instalar y configurar Pacemaker es necesario repetir todos los pasos del Anexo 4.

### **5. CONFIGURACIÓN DE LOS RECURSOS EN PACEMAKER**

Para registrar el nuevo recurso dentro del clúster vamos a modificar el archivo temporal de configuración

/root/config.txt de tal manera que quede de la siguiente manera:

```

node nodoa.mitesis.ger
node nodob.mitesis.ger
node nodoc.mitesis.ger
primitive DNS_DAEMON lsb:named \
    operations $id="DNS_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \
    op monitor interval="15" timeout="15" start-delay="15" \
    meta target-role="started" is-managed="true"
primitive DISK1_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" directory="/disk1/" fstype="ext4" \
    operations $id="DISK1-operations" \
    op start interval="0" timeout="240" \
    op stop interval="0" timeout="100" \
    meta target-role="started" is-managed="true"
primitive DISK1_DRBD ocf:linbit:drbd \
    params drbd_resource="r0" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DNS_IP_ADDR ocf:heartbeat:IPAddr2 \
    params ip="192.168.56.250" cidr_netmask="24" \
    operations $id="DNS_IP_ADDR-operations" \
    op monitor start-delay="0" interval="30" \
    meta target-role="started" is-managed="true"
primitive FTP_DAEMON lsb:vsftpd \
    operations $id="FTP_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \
    op monitor interval="15" timeout="15" start-delay="15" \
    meta target-role="started" is-managed="true"
primitive FTP_IP_ADDR ocf:heartbeat:IPAddr2 \
    params ip="192.168.56.251" cidr_netmask="24" \
    operations $id="FTP_IP_ADDR-operations" \
    op monitor start-delay="0" interval="30" \
    meta target-role="started" is-managed="true"
primitive HTTP_DAEMON lsb:httpd \
    operations $id="HTTP_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \

```

```

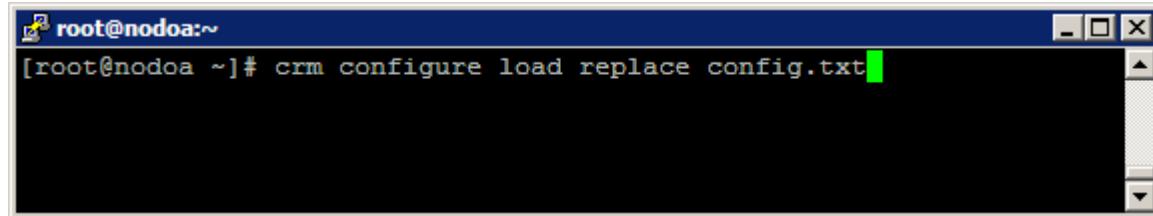
op monitor interval="15" timeout="15" start-delay="15" \
meta target-role="started" is-managed="true"
primitive DISK2_DRBD ocf:linbit:drbd \
    params drbd_resource="r1" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DISK2_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd1" directory="/disk2/" fstype="ext4" \
    operations $id="HTTP_FOLDER-operations" \
    op start interval="0" timeout="240" \
    op stop interval="0" timeout="100" \
    meta target-role="started" is-managed="true"
primitive HTTP_IP_ADDR ocf:heartbeat:IPaddr2 \
    params ip="192.168.56.252" cidr_netmask="24" \
    operations $id="HTTP_IP_ADDR-operations" \
    op monitor start-delay="0" interval="30" \
    meta target-role="started" is-managed="true"
group DISK1_DRBD_GROUP DISK1_DRBD
group DNS_AND_FTP_SERVICES DISK1_FOLDER DNS_IP_ADDR DNS_DAEMON
FTP_IP_ADDR FTP_DAEMON
group DISK2_DRBD_GROUP DISK2_DRBD
group HTTP_SERVICES DISK2_FOLDER HTTP_IP_ADDR HTTP_DAEMON
ms MS_DISK1_DRBD_GROUP DISK1_DRBD_GROUP \
    meta master-max="1" master-node-max="1" clone-max="2" clone-
node-max="1" notify="true"
ms MS_DISK2_DRBD_GROUP DISK2_DRBD_GROUP \
    meta master-max="1" master-node-max="1" clone-max="2" clone-
node-max="1" notify="true"
location LOC_DISK1_FOLDER DISK1_FOLDER -inf: nodoc.mitesis.ger
location LOC_DNS_IP_ADDR DNS_IP_ADDR -inf: nodoc.mitesis.ger
location LOC_FTP_IP_ADDR FTP_IP_ADDR -inf: nodoc.mitesis.ger
location LOC_DISK2_FOLDER DISK2_FOLDER -inf: nodob.mitesis.ger
location LOC_HTTP_IP_ADDR HTTP_IP_ADDR -inf: nodob.mitesis.ger
location LOC_MS_DISK1_DRBD_GROUP MS_DISK1_DRBD_GROUP -inf:
nodoc.mitesis.ger
location LOC_MS_DISK2_DRBD_GROUP MS_DISK2_DRBD_GROUP -inf:
nodob.mitesis.ger
colocation COL_DISK1_DRBD_GROUP inf: DNS_AND_FTP_SERVICES
MS_DISK1_DRBD_GROUP:Master
colocation COL_DISK2_DRBD_GROUP inf: HTTP_SERVICES
MS_DISK2_DRBD_GROUP:Master
order RUN_DNS_AND_FTP_AFTER_DISK1_DRBD_GROUP inf:
MS_DISK1_DRBD_GROUP:promote DNS_AND_FTP_SERVICES:start
order RUN_HTTP_AFTER_DISK2_DRBD_GROUP inf: MS_DISK2_DRBD_GROUP:promote
HTTP_SERVICES:start

```

```
property $id="cib-bootstrap-options" \
    expected-quorum-votes="3" \
    stonith-enabled="false" \
    dc-version="1.1.6-3.el6-
a02c0f19a00c1eb2527ad38f146ebc0834814558" \
    no-quorum-policy="ignore" \
    cluster-infrastructure="openais" \
    last-lrm-refresh="1336327391"
rsc_defaults $id="rsc-options" \
    resource-stickiness="100"
```

Ahora hay que ejecutar el siguiente comando para cargar el archivo de configuración en Pacemaker.

```
crm configure load replace config.txt
```



```
root@nodoa:~#
node nodoa.mitesis.ger \
    attributes standby="off"
node nodob.mitesis.ger
node nodoc.mitesis.ger
primitive DISK1_DRBD ocf:linbit:drbd \
    params drbd_resource="r0" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DISK1_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd0" directory="/disk1/" fstype="ext4"
\
operations $id="DISK1-operations" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="100" \
meta target-role="started" is-managed="true"
primitive DISK2_DRBD ocf:linbit:drbd \
    params drbd_resource="r1" \
    op start interval="0" timeout="240s" \
    op stop interval="0" timeout="100s" \
    meta is-managed="true" target-role="started"
primitive DISK2_FOLDER ocf:heartbeat:Filesystem \
    params device="/dev/drbd1" directory="/disk2/" fstype="ext4"
\
operations $id="HTTP_FOLDER-operations" \
op start interval="0" timeout="240" \
op stop interval="0" timeout="100" \
meta target-role="started" is-managed="true"
primitive DNS_DAEMON lsb:named \
    operations $id="DNS_DAEMON-operations" \
    op start interval="0" timeout="15" \
    op stop interval="0" timeout="15" \
    op monitor interval="15" timeout="15" start-delay="15" \
:
```

## **ANEXO 9 INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR HTTP**

### **1. INSTALACIÓN DEL SERVIDOR HTTP**

Para instalar Apache como servidor de Web es necesario ejecutar el siguiente comando:

```
yum -y install httpd
```

**Repetir el mismo procedimiento en el nodo A**

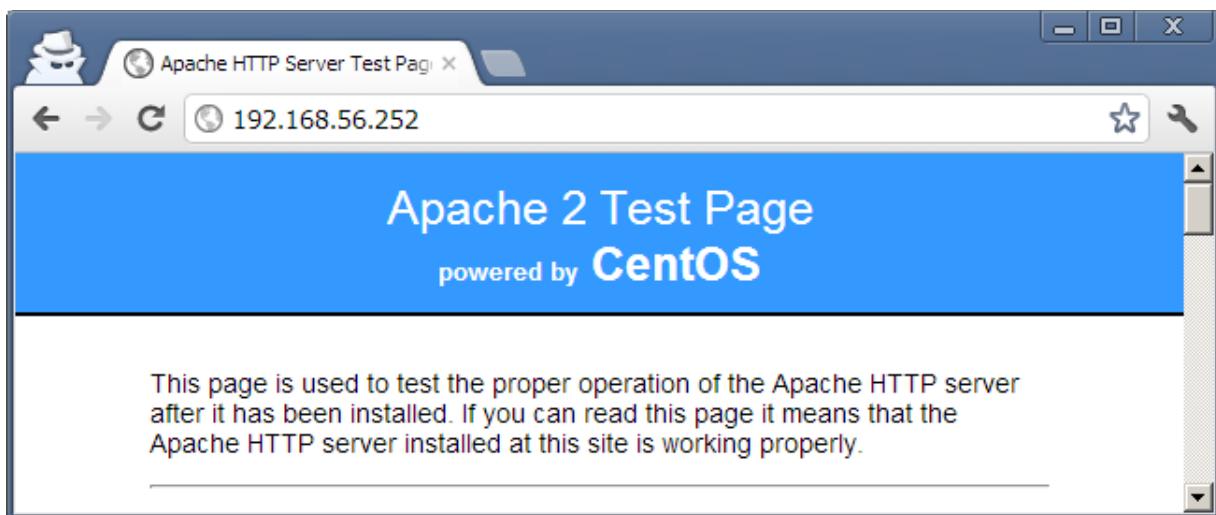
### **2. CONFIGURACIÓN DEL APACHE**

Para poder ejecutar Apache sin errores es necesario agregar la siguiente línea en archivo /etc/httpd/conf/httpd.conf:

```
ServerName www.mitesis.ger:80
```

### **3. PRUEBA DE APACHE**

Para verificar que el servidor web está funcionando escribimos la dirección IP del servidor en un navegador.

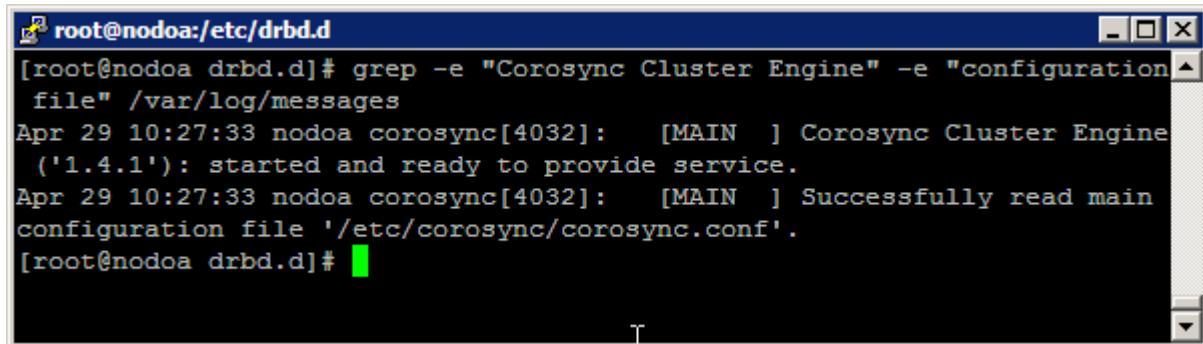


## ANEXO 10 PRUEBAS FINALES DEL CLÚSTER CON 3 NODOS

### 1. PRUEBAS DEL CLÚSTER CON 3 NODOS

Para verificar que tanto la instalación como la configuración del nuevo nodo se han realizado correctamente ejecutamos los siguientes comandos:

```
grep -e "Corosync Cluster Engine" -e "configuration file"  
/var/log/messages
```



```
[root@nodoa drbd.d]# grep -e "Corosync Cluster Engine" -e "configuration file" /var/log/messages  
Apr 29 10:27:33 nodoa corosync[4032]: [MAIN ] Corosync Cluster Engine ('1.4.1'): started and ready to provide service.  
Apr 29 10:27:33 nodoa corosync[4032]: [MAIN ] Successfully read main configuration file '/etc/corosync/corosync.conf'.  
[root@nodoa drbd.d]#
```

```
grep TOTEM /var/log/messages
```

```
root@nodoa:/etc/drbd.d
[root@nodoa drbd.d]# grep TOTEM /var/log/messages
Apr 29 10:27:33 nodoa corosync[4032]: [TOTEM ] Initializing transport
(UDP/IP Multicast).
Apr 29 10:27:33 nodoa corosync[4032]: [TOTEM ] Initializing transmit/r
eceive security: libtomcrypt SOBER128/SHA1HMAC (mode 0).
Apr 29 10:27:33 nodoa corosync[4032]: [TOTEM ] The network interface [
192.168.56.201] is now up.
Apr 29 10:27:34 nodoa corosync[4032]: [TOTEM ] Process pause detected
for 732 ms, flushing membership messages.
Apr 29 10:27:34 nodoa corosync[4032]: [TOTEM ] A processor joined or l
eft the membership and a new membership was formed.
[root@nodoa drbd.d]#
```

```
cat /proc/drbd
```

```
root@nodoa:~
[root@nodoa ~]# cat /proc/drbd
version: 8.4.1 (api:1/proto:86-100)
GIT-hash: 91b4c048c1a0e06777b5f65d312b38d47abaea80 build by phil@Build6
4R6, 2012-04-17 11:28:08
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
    ns:112 nr:0 dw:128 dr:17557 al:4 bm:4 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b
oos:0
1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
    ns:0 nr:4 dw:4 dr:964 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0
[root@nodoa ~]#
```

```
crm_mon -1
```

```
root@nodoc:/var/www/html
=====
Last updated: Mon May  7 20:02:51 2012
Last change: Mon May  7 17:28:57 2012 via crm_resource on nodoc.mitesis.ger
Stack: openais
Current DC: nodoc.mitesis.ger - partition with quorum
Version: 1.1.6-3.el6-a02c0f19a00c1eb2527ad38f146ebc0834814558
3 Nodes configured, 3 expected votes
12 Resources configured.

=====
Online: [ nodob.mitesis.ger nodoa.mitesis.ger nodoc.mitesis.ger ]

Resource Group: DNS_AND_FTP_SERVICES
    DISK1_FOLDER      (ocf::heartbeat:Filesystem):     Started nodob.mitesis.ger
    DNS_IP_ADDR       (ocf::heartbeat:IPaddr2):       Started nodob.mitesis.ger
    DNS_DAEMON        (lsb:named):       Started nodob.mitesis.ger
    FTP_IP_ADDR       (ocf::heartbeat:IPaddr2):       Started nodob.mitesis.ger
    FTP_DAEMON        (lsb:vsftpd):       Started nodob.mitesis.ger
Resource Group: HTTP_SERVICES
    DISK2_FOLDER      (ocf::heartbeat:Filesystem):     Started nodoc.mitesis.ger
    HTTP_IP_ADDR      (ocf::heartbeat:IPaddr2):       Started nodoc.mitesis.ger
    HTTP_DAEMON        (lsb:httpd):       Started nodoc.mitesis.ger
Master/Slave Set: MS_DISK1_DRBD_GROUP [DISK1_DRBD_GROUP]
    Masters: [ nodob.mitesis.ger ]
    Slaves: [ nodoa.mitesis.ger ]
Master/Slave Set: MS_DISK2_DRBD_GROUP [DISK2_DRBD_GROUP]
    Masters: [ nodoc.mitesis.ger ]
    Slaves: [ nodoa.mitesis.ger ]
    FTP_DAEMON_monitor_0 (node=nodoc.mitesis.ger, call=73, rc=5, status=complete)
```

## ANEXO 11 COMANDOS ÚTILES

Corosync dispone de una gran cantidad de utilidades para la administración del clúster que no requieren interrumpir temporalmente su funcionamiento. Aquí se listan algunas de las funcionalidades más básicas:

Comprobar qué servicios hay en un cluster:

```
crm resource list
```

Comprobar en qué nodo se está ejecutando el recurso:

```
crm resource status <id_recurso>
```

Arrancar un recurso:

```
crm resource start <id_recurso>
```

Parar un recurso:

```
crm resource stop <id_recurso>
```

Migrar un recurso a otro nodo:

```
crm resource move <id_recurso> <nodo_destino>
```

Limpiar registro de error de un nodo recurso:

```
crm resource cleanup
```

Dejar un nodo en Standby:

```
crm node standby <nombre_nodo>
```

Poner un Nodo OnLine:

```
crm node online <nombre_nodo>
```

El número de utilidades que ofrece Corosync es muy amplio, pero el detalle de todos y cada uno de ellos, está fuera del alcance este documento sin embargo se recomienda revisar estas herramientas para obtener experiencia en la administración del clúster.

Algunas de las utilidades recomendadas son:

```
crm, cibadmin, crmadmin, crm_attribute, crm_diff, crm_failcount,  
crm_master, crm_mon, crm_resource, crm_standby, crm_uuid y crm_verify.
```

**TABLA 6. PARTICIONES DE DISCO**

<b>Partición</b>	<b>Punto de Montaje</b>	<b>Tamaño</b>	<b>Tipo</b>
/dev/sda1	/	10GB	Ext4
/dev/sda2	No Aplicable	4GB	Swap
/dev/sda3	/usr	40GB	Ext4
/dev/sda4	/var	40GB	Ext4

**TABLA 7. CONFIGURACIÓN DE RED DEL NODO A**

<b>Nodo A</b>	
<b>Parámetro</b>	<b>Valor</b>
Dirección IP	192.168.56.201
Mascara de Red	255.255.255.0
Puerta de Enlace	192.168.56.254
Servidores DNS	192.168.56.250

**TABLA 8. CONFIGURACIÓN DE RED DEL NODO B**

<b>Nodo B</b>	
<b>Parámetro</b>	<b>Valor</b>
Dirección IP	192.168.56.2
Mascara de Red	255.255.255.0
Puerta de Enlace	192.168.56.254
Servidores DNS	192.168.56.250

**TABLA 9. CONFIGURACIÓN DE RED DEL NODO C**

<b>Nodo C</b>	
<b>Parámetro</b>	<b>Valor</b>
Dirección IP	192.168.56.3
Mascara de Red	255.255.255.0
Puerta de Enlace	192.168.56.254
Servidores DNS	192.168.56.250