

UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”

**PROPUESTA DE ALTA DISPONIBILIDAD PARA EL SISTEMA DE
TELEFONÍA IP DEL “INSTITUTO DE PREVISIÓN Y ASISTENCIA
SOCIAL PARA LOS TRABAJADORES DEL MINISTERIO DEL PODER
POPULAR PARA LA EDUCACIÓN” (IPASME).
CASO: SEDE ADMINISTRATIVA PRINCIPAL**

BARQUISIMETO, OCTUBRE 2011

UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGÍA
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

**PROPUESTA DE ALTA DISPONIBILIDAD PARA EL SISTEMA DE
TELEFONÍA IP DEL “INSTITUTO DE PREVISIÓN Y ASISTENCIA
SOCIAL PARA LOS TRABAJADORES DEL MINISTERIO DEL PODER
POPULAR PARA LA EDUCACIÓN” (IPASME).
CASO: SEDE ADMINISTRATIVA PRINCIPAL**

Trabajo presentado como requisito parcial para optar al grado de:
Magister Scientiarum en Ciencias de la Computación

Autor(a) : Ing. Aimará Vargas

Tutor(a) : Prof. Euvis Piña

BARQUISIMETO, OCTUBRE 2011

DEDICATORIA

A Dios y mi Divina Pastora, por llevarme siempre de la mano y bajo su manto, porque nunca me han dejado caer a pesar de los obstáculos.

A mis padres, por ser ejemplo de trabajo, constancia y amor.

A Inti Amarú, mi DIOS del Sol, mi amado hermano, porque sólo tú pudiste llevarme alcanzar esta meta a pesar de haber sufrido tu terrible partida en el camino.

A toda mi familia Vargas y Moya, porque son el motor que siempre me impulsa a seguir adelante. Este logro es también de ustedes.

A mis ahijados, Carlos David, Andrés Alejandro y Bárbara Alondra, para que esta meta lograda sirva de ejemplo en su futuro desarrollo personal y profesional.

A mis grandes amigos, a mis colegas y compañeros en este proyecto, que hoy se consolida, porque todos ustedes de alguna manera contribuyeron a mi desarrollo personal y profesional durante este proceso.

A todas y cada una de las personas que han formado parte de mi vida durante este tiempo, porque de alguna u otra manera seguramente contribuyeron a la consecución de esta ansiada meta. Porque nadie llega a nuestra vida por casualidad...

AGRADECIMIENTO

A la Universidad Centroccidental Lisandro Alvarado, por ser mi Alma Mater, la cual me ha formado como una profesional integral y competitiva dentro de este difícil y cambiante mundo en el cual tenemos que desenvolvemos día a día.

Al personal de la Coordinación de Redes y Comunicaciones de la sede administrativa principal del IPASME, por todo su apoyo y colaboración en la elaboración de este trabajo.

A mi tutora, Ing. Euvís Piña, porque más que una tutora ha sido una compañera paciente y perseverante, trabajando a mi lado hombro con hombro para lograr el cumplimiento de esta meta.

A los profesores Glennys Clemant, Luzneida Matute y William Polanco, por su apoyo y orientación en la primera fase de este trabajo.

A Rafael Valero, Junior Escalona y la Prof. Maritza Torres, por su apoyo y orientación durante el desarrollo de esta investigación.

A todos ustedes, gracias por su apoyo.

INDICE

	PAG.
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE DE CUADROS	vii
ÍNDICE DE FIGURAS	ix
RESUMEN	xi
INTRODUCCIÓN	1
EL PROBLEMA	3
Planteamiento del problema	3
Objetivos	7
<i>General</i>	7
<i>Específicos</i>	7
Justificación e Importancia	8
Alcance y Limitaciones	9
MARCO TEÓRICO	10
Antecedentes	10
Bases Teóricas	12
Telefonía IP	13
Ventajas de un Sistema de Telefonía IP	14
Calidad de Servicio en un Sistema de Telefonía IP	15
Arquitectura de un Sistema de Telefonía IP	17
El software Asterisk	22
Sistemas de Alta Disponibilidad	28
Marco de análisis para determinar los requisitos de Alta Disponibilidad	29
Métodos para garantizar Alta Disponibilidad	31
Software de Alta Disponibilidad	40
Proyectos de Alta Disponibilidad	42
Esquema General de Alta Disponibilidad (Guadamuz, E., 2007)	44
Soluciones de Alta Disponibilidad orientadas a Telefonía IP	46
Identificación de la Institución	49
Bases Legales	52
Constitución de la República Bolivariana de Venezuela	53
Ley Orgánica de Telecomunicaciones	53
Plan de Desarrollo Económico y Social de la Nación 2007-2013	54
Decreto Presidencial 3.390	54
Política Institucional del IPASME	55
Operacionalización de las Variables	55
MARCO METODOLÓGICO	58

Tipo de Investigación	58
Fases del Estudio.....	59
Fase Diagnóstica	59
<i>Población y Muestra</i>	<i>60</i>
<i>Procedimiento Diagnóstico</i>	<i>64</i>
<i>Técnicas e Instrumentos de Recolección de Datos.....</i>	<i>64</i>
<i>Validación de los Instrumentos</i>	<i>66</i>
<i>Determinación de la confiabilidad de los Instrumentos</i>	<i>67</i>
<i>Técnica para el Análisis de los Datos</i>	<i>68</i>
<i>Análisis de los Resultados</i>	<i>69</i>
<i>Conclusiones del diagnóstico</i>	<i>77</i>
<i>Recomendaciones del diagnóstico</i>	<i>77</i>
Estudio de Factibilidad.....	78
Factibilidad Técnica.....	79
Factibilidad Operativa.....	80
Factibilidad Institucional	81
Factibilidad Económica	81
PROPUESTA DEL ESTUDIO.....	83
Esquemas de Alta Disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME.....	83
Descripción de la propuesta.....	92
Arquitectura del Hardware	92
Arquitectura del Software.....	93
Evaluación del diseño propuesto de Alta Disponibilidad.....	121
CONCLUSIONES Y RECOMENDACIONES	138
REFERENCIAS BIBLIOGRÁFICAS	140
ANEXOS	142
ANEXO A . CURRICULUM VITAE DEL AUTOR.....	143
ANEXO B. CUESTIONARIO 01	144
ANEXO C. CUESTIONARIO 02.....	145
ANEXO D. TABLA DE RECOPIACIÓN DE DATOS.....	146
ANEXO E. FORMATOS DE VALIDACION DE LOS INSTRUMENTOS.....	147
ANEXO F. CALCULO DE LA CONFIABILIDAD DE LOS INSTRUMENTOS MEDIANTE EL ALPHA DE CRONBACH.....	148

ÍNDICE DE CUADROS

	PÁG.
Cuadro 1. Umbrales de Calidad de Servicio para VoIP	16
Cuadro 2. Códecs más comúnmente utilizados el Telefonía	17
Cuadro 3. Guía de requisitos del sistema	25
Cuadro 4. Disponibilidad de un sistema 24x7 y tiempos de caída permitidos	32
Cuadro 5. Operacionalización de la variable	57
Cuadro 6. Cálculo de la muestra	63
Cuadro 7. Distribución total de la población y la muestra	63
Cuadro 8. Distribución de frecuencia sobre la opinión de los usuarios con respecto a la fiabilidad del Sistema de Telefonía IP	70
Cuadro 9. Distribución de frecuencia sobre la opinión de los usuarios con respecto a la recuperación ante fallos del Sistema de Telefonía IP.	71
Cuadro 10. Distribución de frecuencia sobre la opinión de los usuarios con respecto a la detección de errores del Sistema de Telefonía IP.	71
Cuadro 11. Distribución de frecuencia sobre la opinión de los usuarios con respecto al impacto del Sistema de Telefonía IP en las actividades laborales	72
Cuadro 12. Distribución de frecuencia sobre la opinión de los usuarios expertos en cuanto a la existencia de Acuerdos de Nivel de Servicio para el Sistema de Telefonía IP.	73
Cuadro 13. Distribución de frecuencia sobre la opinión de los usuarios expertos con respecto a la administración del Sistema de Telefonía IP	74
Cuadro 14. Distribución de frecuencia sobre la opinión de los usuarios expertos con respecto a la redundancia y respaldo de recursos críticos del Sistema de Telefonía IP	75

Cuadro 15. Distribución de frecuencia sobre la opinión de los usuarios expertos con respecto a la estandarización del Sistema de Telefonía IP	76
Cuadro 16. Ventajas y desventajas de los distintos proyectos de Alta disponibilidad de servicios	86
Cuadro 17. Ventajas y desventajas de los distintos proyectos de Alta disponibilidad de datos	89
Cuadro 18. Arquitectura de hardware para cada uno de los nodos del clúster	92

ÍNDICE DE FIGURAS

	PÁG.
Figura 1. Adaptador analógico para interconexión con la Red Telefónica Tradicional	18
Figura 2. Adaptador de Teléfono Analógico (ATA)	18
Figura 3. Softphone X-lite	19
Figura 4. Teléfono IP de gama baja	20
Figura 5. Teléfono IP de gama media	20
Figura 6. Teléfono IP de gama avanzada	21
Figura 7. Tarjeta de comunicaciones para conectar hasta cuatro líneas analógicas	23
Figura 8. La Arquitectura de Asterisk	25
Figura 9. Propiedades de la virtualización: Particionamiento	36
Figura 10. Propiedades de la virtualización: Aislamiento	36
Figura 11. Propiedades de la virtualización: Encapsulación	37
Figura 12. a) Cluster de dos nodos redundantes. b) El servicio se mantiene disponible aún cuando uno de los nodos deja de operar.	46
Figura 13. Arquitectura básica de un Sistema de Telefonía IP	47
Figura 14. Organigrama funcional de la institución	51
Figura 15. Estructura Organizativa de OCTSI	52
Figura 16. Arquitectura actual del Sistema de Telefonía IP de la sede administrativa principal de IPASME	84
Figura 17. Esquema 1. Propuesta básica de Alta disponibilidad en un sistema de Telefonía IP	85
Figura 18. Esquema 2. Propuesta de Alta disponibilidad de servicios integrada con alta disponibilidad de datos	88
Figura 19. Esquema 3. Propuesta integrada de Alta disponibilidad en servicios y datos para el Sistema de Telefonía IP de la sede administrativa principal del IPASME	91

Figura 20. Salida mostrada por el comando cat /proc/drbd indicando que ambos nodos están conectados y sincronizados	100
Figura 21. Arquitectura del Proyecto LINUX-HA	102
Figura 22. Menús disponibles para trabajar con crm	113
Figura 23. Arquitectura del escenario de pruebas	122
Figura 24. Configuración de hardware de la máquina virtual debian1	123
Figura 25. Configuración de hardware de la máquina virtual debian2	124
Figura 26. Configuración del fichero /etc/network/interfaces para ast1	125
Figura 27. Configuración del fichero /etc/network/interfaces para ast2	125
Figura 28. Estructura de particiones de disco de ast1	126
Figura 29. Estructura de particiones de disco de ast1	127
Figura 30. Configuración del fichero /etc/drbd.conf en ambos nodos	128
Figura 31. Inicio del servicio de heartbeat en ast1	131
Figura 32. Inicio del servicio de heartbeat en ast2	132
Figura 33. configuración del fichero cib shadow	133
Figura 34. estatus del cluster antes del apagado de ast1	134
Figura 35. Estatus del cluster luego del apagado de ast1	135
Figura 36. Configuración de red de ast2 luego del apagado de ast1	136
Figura 37. Estatus del cluster luego de la parada del servicio Asterisk en ast2	137
Figura 38. Estatus del cluster luego de la parada del servicio heartbeat en ast2	138

UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGÍA
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

**PROPUESTA DE ALTA DISPONIBILIDAD PARA EL SISTEMA DE
TEFEFONÍA IP DEL “INSTITUTO DE PREVISIÓN Y ASISTENCIA
SOCIAL PARA LOS TRABAJADORES DEL MINISTERIO DEL PODER
POPULAR PARA LA EDUCACIÓN” (IPASME).
CASO: SEDE ADMINISTRATIVA PRINCIPAL**

Autor(a): Aimará Vargas

Tutor(a): Euvis Piña

RESUMEN

La presente investigación tuvo como objetivo fundamental diseñar una propuesta de Alta Disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del Instituto de Previsión y Asistencia Social para los Trabajadores del Ministerio del Poder Popular para la Educación” (IPASME). Este requerimiento surge por el volumen creciente de usuarios dentro de la Central de Telefonía IP basada en software libre con la que cuenta actualmente este instituto, lo que genera la necesidad de garantizar la disponibilidad permanente del servicio. En virtud de que el estudio estuvo enmarcado dentro de la modalidad de Proyecto Factible, se llevó a cabo en tres (3) fases: fase I de diagnóstico, fase II de factibilidad técnica, operativa, económica e institucional del diseño propuesto y fase III en la que se plantea el Diseño de la propuesta de Alta Disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME. Dado que la sede principal de la institución, objeto de estudio en este trabajo, tiene un alto impacto en las operaciones de la institución a nivel nacional, atendiendo los requerimientos de aproximadamente 1.500.000 personas al año, resulta de vital importancia la disponibilidad garantizada de sus servicios en general, y más aun de sus servicios de comunicaciones.

Palabras clave: Alta Disponibilidad, Sistema de Telefonía IP, Disponibilidad de Servicio, Servicios de Comunicaciones.

INTRODUCCIÓN

Durante las últimas décadas, los avances en materia de nuevas tecnologías de información y comunicaciones han experimentado un crecimiento exponencial, tanto a nivel de implantación de nuevas tecnologías como a nivel de servicios y funcionalidades de los mismos. Particularmente, los sistemas de telefonía tradicionales han experimentado una importante evolución donde las redes de conmutación de circuitos están siendo rápidamente desplazadas por las redes de conmutación de paquetes. Estas últimas mejor conocidas como redes IP (por las siglas en inglés de su protocolo base: El Protocolo de Internet).

Los nuevos sistemas de Telefonía IP, además de prestar los servicios básicos de comunicaciones, incorporan una gama de atractivas prestaciones tales como: identificador de llamadas, transferencia de llamadas, conferencias, parqueo, entre otros. Sin embargo, uno de los retos que debe superar para poder reemplazar definitivamente a las redes telefónicas tradicionales lo constituye el paradigma de la alta disponibilidad, lo cual supone que el tiempo efectivo del servicio debe estar en el orden del 99.999 % (indicador para la disponibilidad del servicio en la telefonía tradicional).

En este sentido, se hace indispensable incorporar el concepto de alta disponibilidad a los Sistemas de Telefonía IP implantados ó propuestos en las organizaciones, lo que significa realizar un estudio particular que va desde el establecimiento de un Acuerdo de Nivel de Servicio (definiendo cuánto tiempo y en que horarios debe estar disponible el mismo), pasando por la evaluación análisis de los distintos métodos existentes para garantizar la alta disponibilidad, hasta llegar a la implementación en los Sistemas de Alta Disponibilidad del servicio específico para la organización a la cual servirá.

De lo anterior, resulta oportuno el estudio de la incorporación de alta disponibilidad al Sistema de Telefonía IP del Instituto de Previsión y Asistencia Social para los Trabajadores del Ministerio del Poder Popular para la Educación” (IPASME), más específicamente al Sistema de Telefonía IP de la sede administrativa principal de esta institución .

Actualmente esta institución pública cuenta con un sistema de Telefonía IP híbrido, conformado por una central propietaria y una central basada en software libre. Es en esta última central donde se han creado los nuevos usuarios para no incurrir en gastos adicionales de licencias originando que día a día esta central esté más comprometida en la prestación ininterrumpida del servicio.

El presente trabajo de investigación está estructurado en cinco (5) capítulos a saber:

El capítulo I, denominado El Problema, donde se presenta el planteamiento del problema, los objetivos que fueron planteados tanto a nivel general como específico, justificación, importancia y alcance de la investigación.

En capítulo II, denominado Marco Teórico, en el cual se desarrollan los antecedentes más relevantes de la investigación, seguidos de las bases teóricas y legales que la sustentan.

El capítulo III, denominado Marco Metodológico, donde se detalla la naturaleza y diseño de la investigación. Además se desarrollan la dos primeras fases del estudio factible: Fase de diagnóstico y Fase de Factibilidad.

El capítulo IV, denominado Propuesta del Estudio, el cual muestra y explica el diseño propuesto. Así como los resultados obtenidos en las pruebas realizadas.

El capítulo V, denominado Conclusiones y Recomendaciones, donde se indican brevemente los aportes del trabajo realizado y las sugerencias para mejorar el producto obtenido.

CAPÍTULO I

EL PROBLEMA

Planteamiento del problema

Los sistemas de telefonía representan hoy en día, una de las áreas de mayor evolución en respuesta a las necesidades que establece la población. Esta demanda ha traído consigo la búsqueda y prueba de nuevas tecnologías telefónicas orientadas a bajos costos, altas prestaciones y extensas funciones.

Una de las tecnologías que surgió para impulsar estos cambios es la Voz sobre IP (VoIP) la cual es definida por Sierra, A. (2008) como: “un conjunto de protocolos para transporte de voz sobre redes IP” (Pág.1). Esta sirve de base para lo que se conoce como Telefonía IP, cuya definición según el mismo autor es: “un sistema avanzado de comunicaciones que utiliza el protocolo de Internet como medio de transporte para crear un sistema telefónico con todas las funciones de la telefonía tradicional, pero que además agrega nuevas posibilidades.” (Pág. 1).

Las soluciones de Telefonía IP traen consigo además de la reducción de los costos de las llamadas, servicios tales como el correo de voz, conferencias, simplificación de la administración al integrar los sistemas de voz y datos, entre otros.

Tal y como lo señala la empresa 3CX (2009), desde su sitio en Internet:

Telefonía IP es sin lugar a dudas, uno de los desarrollos tecnológicos que está siendo rápidamente adoptado por muchas empresas hoy en día. Una de las principales razones de esta rápida migración a telefonía Internet es que hace mucho más fácil la integración de todos los medios de comunicación, dispositivos de comunicación y medios. De esta manera los usuarios pueden estar en contacto con cualquiera, desde cualquier parte que ellos estén y en tiempo real. En resumen, la telefonía IP permite que las *comunicaciones unificadas* sean parte del ambiente de la empresa,

ayudando a las empresas a ahorrar dinero e incrementar la productividad de los empleados.

Dado que las soluciones de Telefonía IP se han convertido en una alternativa para usuarios y organizaciones a nivel mundial y a gran escala, se ha planteado el reto de la optimización del servicio, no sólo a nivel de la calidad del tráfico de la voz por el mismo medio de datos, sino que además se hace indispensable garantizar la alta disponibilidad del sistema de comunicación y optimizar el rendimiento, aprovechando al máximo los recursos de hardware con los que se cuenta.

Resulta oportuno mencionar una definición de Alta Disponibilidad enunciada por Colobran, M. y Otros (2008): “La alta disponibilidad de un sistema consiste en reducir al mínimo la posibilidad de un error de hardware o un defecto de software que origine la interrupción de uso del sistema o la pérdida de datos del sistema”. (Pág. 78)

De aquí, el reto de garantizar la disponibilidad de un sistema de Telefonía IP, comprende estudiar todos aquellos factores que puedan generar la interrupción de uno o más servicios que lo componen, los cuales pueden ir desde la interrupción de la alimentación eléctrica, la caída del servicio de Internet o inclusive fallas de hardware o software en el equipo que funciona como central telefónica. En este sentido, Casas, S. y Reinaga, H. (2009), señalan lo siguiente:

El paradigma de la alta disponibilidad (*high availability*) en servicios de telefonía supone un servicio por lo menos con una disponibilidad del 99.999% de tiempo efectivo de servicio (comúnmente llamado *cinco nueves*). A lo largo de la historia de la telefonía tradicional por conmutación de circuitos los fabricantes hicieron de este indicador el índice de disponibilidad del servicio característico de la PSTN (Public Switched Telephone Network). (Pág. 1)

Dado que la Telefonía IP surge como una innovación de las comunicaciones telefónicas que persigue ofrecer más y mejores servicios que la telefonía convencional, es de suponer que un sistema de telefonía IP debe garantizar una

disponibilidad del servicio igual ó mayor que la ofrecida por la telefonía tradicional. Por tanto, se puede pensar en la integración de dos tecnologías: Sistemas de Alta Disponibilidad y Telefonía IP. .

Sin embargo, en base a una investigación preliminar realizada por parte del autor en materia de implantación de sistemas de telefonía IP, se ha constatado que el tema de la alta disponibilidad del servicio telefónico en muchos casos es dejado a un lado. Tal es el caso de la Solución de Voz sobre IP basada en Software Libre para el Instituto de Previsión y Asistencia Social para los Trabajadores del Ministerio del Poder Popular para la Educación (IPASME) propuesta por García, K. (2009), donde el tema de la alta disponibilidad del servicio no fue tomado en cuenta dentro del alcance de dicha investigación.

En la actualidad, la sede administrativa principal del IPASME ubicada en la ciudad de Caracas, cuenta con un Sistema de Telefonía IP híbrido conformado por: una Central IP propietaria HiPath4000 de Siemens la cual maneja un total de 245 extensiones y una Centralita IP de software libre basada en la propuesta de García, K. (2009), que cuenta actualmente con un aproximado de 200 extensiones. Estas centralitas están configuradas de manera que se comunican entre sí por medio de la red IP y se comunican con el mundo exterior por medio de las líneas de telefonía tradicional que se encuentran instaladas en la central Siemens. Dicha central integra todos los servicios de intercambio de información (voz, video y datos) en una única infraestructura de red e inclusive incorpora elementos de alta disponibilidad como parte de sus especificaciones técnicas. Por otro lado, la centralita IP basada en Software libre que maneja el resto de las extensiones logra aplicaciones interesantes como el identificador de llamadas, transferencia de llamadas, conferencia, parqueo, captura, monitoreo, e infinidad de aplicaciones.

Cabe destacar que, a pesar de la gran cantidad de ventajas ofrecidas por la Central IP propietaria HiPath4000 de Siemens, no deja de ser un sistema telefónico complejo que no puede ser modificado fácilmente y que requiere periódicamente de la intervención de un técnico especialista en esa central específica, lo que hace que la

implantación de esta central este casada de por vida con el proveedor de la misma. Además, la Central IP HiPath4000 trabaja bajo el protocolo propietario CorNet, lo que dificulta su conexión con terminales que no utilicen el mismo protocolo. Sumando a esto, se tiene el hecho de que al presentarse el requerimiento de nuevas extensiones, se debe adquirir una nueva licencia para cada nueva extensión requerida.

Por otra parte, a través de un proceso de observación directa y entrevistas no estructuradas realizadas por el investigador al personal encargado del Sistema de Telefonía IP de la sede administrativa principal del IPASME, se determinó que esta institución no cuenta con asignación económica para la adquisición de licencias para las nuevas extensiones de Telefonía IP que son requeridas, haciendo más evidente la necesidad de desarrollar la plataforma de telefonía IP modelada por García (2009).

Por lo tanto, tomando en consideración que en el caso particular de la sede administrativa principal del IPASME, se trata de una edificación donde funcionan todas las oficinas centrales encargadas de coordinar los distintos procesos administrativos a nivel nacional (donde laboran alrededor de 800 trabajadores), se hace indispensable asegurar la alta disponibilidad de este sistema de comunicaciones crítico para la institución.

Esta central IP basada en software libre debe alojar los nuevos requerimientos que surjan en la institución (además de las 200 extensiones que ya maneja) en un computador de escritorio con procesador Intel Core duo y memoria RAM de 3 G, y es el personal técnico de la misma institución el responsable de solventar cualquier situación que se presente, inclusive si llegara a presentarse alguna falla en el equipo donde funciona esta central.

A partir de la situación antes expuesta, surgen las siguientes interrogantes:

¿Hasta que punto las soluciones de Telefonía IP planteadas actualmente en la sede administrativa principal del IPASME garantizan una alta disponibilidad del servicio?

¿Cual es la factibilidad técnica, financiera e institucional de plantear el diseño de un Sistema de Telefonía IP de Alta Disponibilidad para la sede administrativa principal del IPASME?

¿Cuáles serían los posibles escenarios de Alta Disponibilidad aplicables al actual Sistema de Telefonía IP de la sede administrativa principal del IPASME?

¿Como se puede diseñar un Sistema de Telefonía IP de Alta disponibilidad que cubra los requerimientos de la sede administrativa principal del IPASME en cuanto al tiempo efectivo del servicio?

Objetivos

General

Plantear una propuesta de alta disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del “Instituto de Previsión y Asistencia Social para los trabajadores del Ministerio del poder popular para la Educación” (IPASME).

Específicos

1. Diagnosticar la necesidad de la incorporación de alta disponibilidad al Sistema de Telefonía IP de la sede administrativa principal del IPASME.
2. Determinar la factibilidad técnica, operativa, económica e institucional de incorporar alta disponibilidad al Sistema de Telefonía IP de la sede administrativa principal del IPASME.
3. Estudiar los escenarios de Alta Disponibilidad aplicables al actual Sistema de Telefonía IP de la sede administrativa principal del IPASME.
4. Diseñar una propuesta de Alta disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME.

5. Mostrar mediante un escenario de pruebas la Alta Disponibilidad del servicio de Telefonía IP presentado en el diseño propuesto.

Justificación e Importancia

El IPASME es un organismo asistencial dependiente del Ministerio del Poder Popular para la Educación que cuenta con 64 sedes a nivel nacional, divididas en 61 unidades IPASME, 3 sedes administrativas y 2 destinos turísticos. A su vez las unidades IPASME están categorizadas como tipo I, II y III respectivamente, las cuales se diferencian por su tamaño en estructura, la población que abarca y los servicios que presta.

Según información proporcionada al autor del presente proyecto por parte de la Gerencia de Salud Integral y la Gerencia de Afiliación y Contribuciones de este instituto, en la actualidad existe una cantidad aproximada de 250.000 afiliados (activos y jubilados), lo que representa un total aproximado de 1.250.000 beneficiarios considerando un promedio de 5 beneficiarios por cada afiliado. A esto se le suma la cantidad de 25.000 personas, no afiliadas ni beneficiarios, que reciben servicio asistencial de cortesía cada año a nivel nacional.

Esta cantidad de usuarios, el tipo de servicios que presta, el presupuesto que maneja y la magnitud de sus inversiones, hacen del IPASME una institución compleja en materia de Previsión y Asistencia Social. Más aún si se toma en cuenta que la gran mayoría de los procesos administrativos que ejecutan las distintas unidades distribuidas en todo el territorio nacional, deben ser avalados por las oficinas centrales ubicadas en la sede administrativa principal de esta institución, donde la disponibilidad permanente de los servicios de comunicaciones tales como el Sistema de Telefonía IP, se hacen indispensables para asegurar la continuidad de todos estos procesos administrativos.

La caída de todo o parte del sistema de comunicaciones telefónicas instalado en la sede administrativa principal, supone el retardo de los procesos administrativos que se siguen ya que las comunicaciones telefónicas son la herramienta fundamental utilizada desde y hacia las unidades regionales distribuidas en todo el territorio nacional para interactuar de forma rápida con las oficinas centrales ubicadas en la sede administrativa principal. Además, internamente la sede administrativa es una edificación con mas de 800 trabajadores distribuidos en 10 niveles y cuyas funciones, por lo general, están integradas con las funciones de personal de otras oficinas internas pero cuya ubicación puede ser distante en ocasiones lo que significaría un mayor tiempo en la ejecución de los procesos, si el personal no dispone internamente del sistema telefónico.

Por estas razones, resulta fundamental para la Oficina de Tecnología y Sistemas de Información, específicamente para la Coordinación de redes y Comunicaciones, tener asegurada la continuidad del servicio telefónico tanto interno como externo y por medio del diseño propuesto en este trabajo, se plantea un escenario donde se garantiza la disponibilidad del servicio telefónico en esta institución.

Alcance y Limitaciones

Se tuvo como meta, plantear el diseño detallado de una propuesta de Alta Disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME y la construcción de un escenario de pruebas basado en máquinas virtuales, donde se muestre el funcionamiento a nivel de software de la propuesta diseñada

CAPÍTULO II

MARCO TEÓRICO

A continuación se presentan los fundamentos teóricos que conllevan al desarrollo del presente trabajo, iniciando con las investigaciones que dieron origen a la temática planteada. Luego se desarrollan los términos más destacados en cuanto a Telefonía IP y Sistemas de Alta disponibilidad, para posteriormente destacar los aspectos legales que de alguna u otra manera pueden influir en el desarrollo de la investigación.

Antecedentes

En cuanto a investigaciones previas referentes a Sistemas de Telefonía IP, Alta Disponibilidad y la integración de ambos, se citan a continuación los trabajos con mayor pertinencia:

En la Universidad Nacional Experimental Politécnica “Antonio José de Sucre”, vice-rectorado “Luis Caballero Mejías”, García, K. (2009), desarrolló el Trabajo de Grado titulado: “Diseño e Implantación de una Solución de Voz sobre IP Basado en Software Libre para las Unidades del IPASME”. El objetivo general planteado es esta investigación fue: Diseñar e implementar una solución Voz sobre IP basada en software libre para las unidades del IPASME.

La investigación se realizó siguiendo la metodología de proyecto factible, realizando un estudio exhaustivo de las distintas opciones de software libre para Voz sobre IP, tanto a nivel de centralitas telefónicas como a nivel de teléfonos IP implementados en software. Posteriormente, se realizó el diseño detallado del Sistema de Voz sobre IP para la sede administrativa de Caracas y se planteó un

diseño más general para las unidades según su tipo (unidades Tipo I, Tipo II ó Tipo III). A nivel de implantación, en este trabajo se realizó la instalación y configuración de Asterisk en un servidor de pruebas con 10 extensiones, ubicado en la Coordinación de Redes de OCTSI en la sede administrativa del IPASME.

Finalmente, en la sección de recomendaciones el autor señala:

Se sugiere que en las sedes de mayor tráfico de voz se instale el sistema con un clúster redundante para que este tenga menos posibilidad de dejar de prestar el servicio, adicionalmente se sugiere que el sistema de comunicaciones posea un alternativa a la alimentación eléctrica a través de un banco de baterías que pueda mantener el servicio activo durante la cantidad de tiempo suficiente.

Este último señalamiento constituye la base fundamental de la presente investigación, ya que busca justamente atender a esta recomendación brindando una solución de alta disponibilidad para el Sistema de Telefonía IP del IPASME.

Por otra parte, en la Escuela de Telecomunicaciones de la Facultad de Ingeniería de la Universidad Católica Andrés Bello, Figueroa, W. (2008) presentó su trabajo especial de Grado: “Implementación de una Central Telefónica Privada de VoIP bajo Trixbox 2.2 de Alta Disponibilidad para la Gerencia de Direccionamiento Tecnológico de CANTV”. Este estudio se desarrolló como una investigación no experimental de tipo aplicada y apoyada en la investigación de campo. El enfoque de esta investigación gira en torno a la construcción del sistema telefónico y el clúster de alta disponibilidad en un escenario de pruebas basado en máquinas virtuales para su posterior implementación en máquinas reales. Esta investigación marca un importante precedente con respecto al presente trabajo, ya que se plantea el uso de máquinas virtuales para los escenarios de pruebas en sistemas de telefonía IP, similar al escenario usado para probar la propuesta diseñada para el IPASME.

Cabe destacar que aún cuando la temática planteada en ambos trabajos está íntimamente ligada no presentan el mismo enfoque, ya que Figueroa, W. (2009), plantea el uso de una Central Telefónica Privada, mientras que en esta investigación

se propone un diseño basado en tecnologías libres. Además, cada organización tiene sus propios requerimientos, por lo que se amerita un estudio, diseño e implantación particular para poder atender las necesidades específicas del entorno en el cual se quiere desarrollar el Sistema de Telefonía IP.

En España, Almería, Méndez, F. (2009) presentó una tesis con el título: “Diseño e Implementación de un Sistema VoIP de Alta Disponibilidad y Alto Rendimiento” cuyo objetivo general fue: Diseñar e Implementar un Sistema VoIP de Alta Disponibilidad y Alto Rendimiento para la Universidad de Almería.

Este estudio se realizó siguiendo una metodología de proyecto factible apoyado en la investigación documental. El trabajo contempló una revisión teórica de los sistemas de Voz IP de Alta Disponibilidad y Alto Rendimiento, y el Diseño, implementación y Prueba del Sistema de VoIP de alta disponibilidad y alto rendimiento en la Universidad de Almería. Los esquemas utilizados en el trabajo referenciado sirvieron de base inicial para los que se plantearon en esta investigación, con la diferencia que el diseño propuesto en el presente trabajo se orientó a soluciones basadas en software para un mayor aprovechamiento de los recursos de hardware.

Bases Teóricas

El presente trabajo de investigación se sustenta fundamentalmente en dos áreas teóricas a estudiar: Telefonía IP y Sistemas de Alta Disponibilidad. En cuanto a Telefonía IP, es importante plantear una definición clara y formal, escribir detalladamente sus ventajas, definir los parámetros que rigen la calidad del servicio y diferenciar los componentes que integran su arquitectura.

Con respecto a los Sistemas de alta disponibilidad, se presenta el marco de análisis para determinar sus requisitos. También se deben describir los métodos existentes para garantizar la alta disponibilidad, así como diferenciar las soluciones orientadas a servicios de aquellas orientadas a datos y presentar un esquema general

de alta disponibilidad, para luego estudiar aquellos esquemas aplicables a Sistemas de Telefonía IP.

Finalmente, se describen los aspectos más importantes que permiten conocer mejor a la institución que servirá como escenario para el desarrollo de la investigación.

Telefonía IP

Ramírez, T. y Díaz, J. (2007) definen la Telefonía IP como:

Una tecnología que permite que las señales de voz viajen a través de Internet empleando el Protocolo de Internet (IP). Esto significa que la señal de voz se envía en forma de paquetes de datos en lugar de enviarla a través de circuitos conmutados utilizando la PSTN. (Pág. 2)

Recopilando varios conceptos enunciados por diversos autores, se puede describir la Telefonía IP como una tecnología que permite integrar en una misma red (basada en el protocolo de Internet -IP) las comunicaciones de voz y datos, orientada a brindar soluciones distribuidas, de fácil expansión e inteligencia a todos los niveles. Muchas veces se utiliza el término de redes convergentes o convergencia IP, aludiendo a un concepto un poco más amplio de integración en la misma red de todo tipo de comunicación (voz, video y datos).

Ventajas de un Sistema de Telefonía IP

Barbéran, J. (2009), describe las ventajas de un Sistema de Telefonía IP en los siguientes términos:

Ahorros de costos

1. Simplificación de la infraestructura: Una única plataforma técnica para voz y datos lo cual se traduce en menor inversión, mantenimiento y formación de la infraestructura.
2. Simplificación del cableado de red: unificación del cableado de voz y datos en Ethernet y posibilidad de compartir un único punto de red entre computador y teléfono.
3. Menores costes de gestión: Las extensiones se pueden reubicar simplemente cambiando los teléfonos IP de sitio y punto de red. Los cambios de configuración se pueden hacer en remoto.
4. Llamadas internas gratis entre sedes de una empresa.
5. Fácil acceso a proveedores VoIP, con llamadas muy económicas y otros servicios avanzados de gran valor pero muy asequibles a cualquier empresa.

Mejoras funcionales

1. Unificación del sistema de Telefonía entre sedes.
2. Plan de numeración integrado.
3. Llamada directa extensión a extensión sin coste.
4. Centralización/Diversificación del puesto de operadora: Las llamadas entrantes pueden dirigirse a cualquier operadora independientemente de su ubicación, siguiendo criterios de disponibilidad o carga de trabajo.
5. Gestión centralizada del sistema de telefonía.

Mejoras en la Atención a los Clientes

1. Música en espera basada en archivos tipo MP3.

2. Operadora Automática (IVR).
3. CTI: Integración con sistemas de gestión (CRM), para llamadas automáticas o identificación datos del cliente por su CLID.
4. ACD: Distribución de llamadas entrantes entre agentes comerciales.
5. IPCC: Aplicaciones específicas para centros de llamadas (Call Centers) o departamentos comerciales.

Incremento en la Productividad del empleado

1. Sistema de correo de voz (Voice Mail) integrado con correo electrónico y directorio.
2. Movilidad: posibilidad de acceder a su extensión telefónica (IP) en cualquier punto.
3. Reubicación en la red corporativa, en cualquier delegación
4. Acceso para Teletrabajo.

Calidad de Servicio en un Sistema de Telefonía IP

Para Rodríguez, R. (2006), QoS (por sus siglas en inglés Quality of Service) es una medición colectiva del nivel de servicio entregado a un usuario e indica que, aunque en una red de Telefonía IP hay muchos parámetros de QoS (ancho de banda, pérdida de paquetes, retraso, jitter, porcentaje de llamadas completadas, tiempo de establecimiento de llamada, entre otros) los factores más importantes son:

1. *Retraso*: el cual es medido de Terminal a Terminal a través de una red. En particular para telefonía basada en conmutación de paquetes, el retraso es medido desde el punto en donde la voz es codificada en la fuente, a través de la red de paquetes y hasta el punto donde es decodificada en el punto destino.
2. *Variación del retraso (jitter)*: un paquete experimenta un cierto retraso y el próximo paquete en la misma conexión a través de los mismos

conmutadores, encuentra diferentes condiciones de tráfico en la red y por lo mismo experimenta un retraso diferente. A esta variación en el retraso se le conoce como jitter, causando distorsión en el extremo final debido a que los paquetes contienen información de audio.

3. *Pérdida de paquetes:* debido a la tecnología de conmutación de paquetes en que está basada el protocolo de Internet, las sesiones establecidas no tienen recursos de transporte asignados de manera exclusiva. Esto trae como consecuencia que se pueda presentar congestión en algunos enrutadores obligándolos a descartar paquetes de manera aleatoria o predeterminada cuando la demanda de tráfico excede el tamaño de sus buffers. Existen umbrales de estos parámetros para diversos niveles de servicio presentando los más recomendados en el cuadro 1.

Cuadro 1.

Umbrales de Calidad de Servicio para VoIP

Parámetro	Unidades	Niveles de Servicio				
		Malo	Pobre	Regular	Bueno	Excelente
Retraso	mseg.	>600	>400	>250	>100	100<=
Jitter	mseg.	>335	>175	>125	>75	75<
Pérdida de paquetes	%	>10	>8	>5	>1	1<=
Llamadas completadas	%	65<	75<	80<	85<	>=85
Llamadas perdidas	%	<20	<15	<13	<10	>=10
Establecimiento de llamada	seg.	>25	>15	>11	>7	<7
Opinión media	MOS	2.8<	3.4<	3.7<	4.2<	>4.2

Fuente: Rodríguez, R. (2006)

Los *códex* son algoritmos para la compresión de la señal de voz. De las características del *códec* va a depender el ancho de banda que se requiere para el flujo de datos y la calidad de la voz. Generalmente, debe hacerse un balance entre estos dos factores: a mayor calidad, mayor ancho de banda requerido, y si el ancho de banda se reduce, la calidad de la voz se ve reducida. Asimismo, cuando la compresión es elevada se requiere un mayor procesamiento para comprimir y descomprimir cada paquete, lo cual hay que considerar cuando se va a implementar un sistema de VoIP

de muchos usuarios. En el cuadro 2 se hace una referencia breve de los códecs más comúnmente utilizados.

Cuadro 2.

Códecs más comúnmente utilizados el Telefonía

<i>Códec</i>	<i>Ancho de Banda</i>	<i>Características</i>
G.711	64 kbps	También es conocido como PCM, es el <i>códec</i> fundamental de la PSTN.
G.726	16, 24 ó 32 kbps	Es un codec adaptivo y no requiere mucho uso de CPU. La calidad es similar al G.711.
G723.1	5.3 ó 6.3 kbps	Es un <i>códec</i> diseñado para bajo uso de ancho de banda. Requiere licenciamiento.
G.729A	8 kbps	Logra una excelente calidad de sonido con poco uso de ancho de banda. Requiere licenciamiento y mucho uso de CPU.
GSM	13 kbps	Utiliza poco ancho de banda y un bajo consumo de CPU. Es de uso libre, pero se considera que su calidad es inferior que los códecs licenciados.
iLBC	13.3 ó 15.2 kbps	Ofrece una buena calidad de voz y utiliza un ancho de banda bajo. Sus algoritmos tienen un alto consumo de recursos del sistema.
Speex	Entre 2.15 y 22.4 kbps	Es un <i>códec</i> capaz de adaptarse dinámicamente a las condiciones de la red.

Fuente: Guadamuz, E. (2007)

Arquitectura de un Sistema de Telefonía IP

En base a la investigación documental realizada por Méndez, F. (2009), la arquitectura de un sistema de Telefonía IP desde el punto de vista de su distribución, puede ser: (a) *centralizada* ó (b) *distribuida*. El enfoque centralizado puede no disponer de la suficiente flexibilidad para adoptar las futuras innovaciones tecnológicas. Por otro lado la arquitectura distribuida es más compleja que la arquitectura centralizada. Sea partiendo de un enfoque u otro, los componentes que forman parte de la arquitectura de un Sistema de Telefonía IP son los siguientes:

Pasarelas ó Gateways: Estas son necesarias si un teléfono tradicional es usado en alguno de los lados de la llamada telefónica. Se trata del enlace con la red telefónica tradicional, actuando de forma transparente para el usuario. Existen pasarelas que actúan a nivel de los servidores de VoIP (Tarjetas y extensiones de interconexión con la Red Telefónica Tradicional. Véase la figura 1) como también se tienen pasarelas a nivel de los clientes (Adaptador de Teléfonos Analógicos ATA, utilizado para conectar un teléfono ordinario a una red de VoIP. Véase la figura 2).



Figura 1. Adaptador analógico para interconexión con la Red Telefónica Tradicional

Fuente: Méndez, F. (2009).



Figura 2. Adaptador de Teléfono Analógico (ATA)

Fuente: Méndez, F. (2009).

Terminales ó Clientes: Son los sustitutos de los actuales teléfonos. Se pueden implementar de dos formas: **(a) software**, también conocidos como softphone, el cual es un software que hace una simulación de teléfono convencional por computadora. Es decir, permite usar la computadora para hacer llamadas a otros softphones o a otros teléfonos convencionales. (Wikipedia, 2010) Existe una variedad de softphones de distribución gratuita, y el más empleado hoy en día es X-Lite, cuya interfaz de usuario puede observarse en la figura 3; **(b) hardware**, que lo conforman los teléfonos IP, equipos especialmente diseñados para conectarse a una red de Telefonía IP. Según sus características y funcionalidades los Teléfonos IP pueden ubicarse en tres categorías: gama baja, gama media y gama avanzada, tal y como se pueden apreciar en la figura 4, la figura 5 y a figura 6 respectivamente.



Figura 3. Softphone X-lite

Fuente: Méndez, F. (2009).



Figura 4. Teléfono IP de gama baja

Fuente: Méndez, F. (2009).



Figura 5. Teléfono IP de gama media

Fuente: Méndez, F. (2009).



Figura 6. Teléfono IP de gama avanzada

Fuente: Méndez, F. (2009).

Red IP: Las redes IP, ya sean de datos o de voz, son diseñadas en base a muchas variables. Dos factores importantes que se necesitan considerar en el diseño de redes son: servicio y costo. El servicio es esencial para mantener la satisfacción del cliente, y el costo es siempre un factor en el mantenimiento de la red misma. Una forma de mantener estos aspectos en un intervalo aceptable es por medio de la utilización óptima de los canales de comunicación a través de la ingeniería de tráfico.

Centralita IP: “Provee las aplicaciones y herramientas para soportar el enrutamiento de llamadas a través de una red IP, pudiendo ser esta una red privada, una Intranet ó Internet” (Figueroa, 2008). Son el centro de todo sistema de telefonía IP y entre las múltiples operaciones que pueden manejar (validación de usuarios, contabilidad, tasación, recolección y distribución de utilidades, enrutamiento, administración general del servicio, carga de clientes, registro de usuarios, entre otros) también manejan las operaciones realizadas por las centrales telefónicas convencionales.

A nivel de hardware, pueden implementarse de dos formas: **(a) PBX IP**, que son equipos diseñados por un fabricante específico sólo para cumplir funciones de centrales telefónicas. De requerir actualizaciones o expansiones a nivel de hardware, por lo general tienen exclusividad con el fabricante original. **(b) Servidores**, que son

equipos de computación robustos utilizando software especializado para ofrecer diversos servicios entre los cuales figura el servicio de Telefonía IP. Los servidores pueden estar dotados de hardware adicional que no tiene que ser provisto necesariamente por la misma empresa fabricante.

En cuanto al software mediante el cual funcionan las centralitas IP, se pueden distinguir dos grandes grupos: **(a) Propietarios**, mayormente utilizados por las PBX IP, cuyos fabricantes del hardware limitan el funcionamiento de sus PBX a un único software previsto por ellos mismos. Tal es el caso de las PBX IP fabricadas por las empresas Cisco, Panasonic, 3Com, entre otras. **(b) Gratuitos**, los cuales son implementados en centralitas IP de tipo servidor y tienen como base el principal software de centralita IP que hasta ahora se conoce: Asterisk. Por ser este último el software con mayor influencia en la mayoría de soluciones de telefonía IP hoy en día, a continuación se describen sus principales características según Guadamuz, E. (2007).

El software Asterisk

Asterisk es un software de código abierto que implementa una central telefónica IP, diseñado originalmente para que se ejecute sobre el sistema operativo GNU/Linux. Como cualquier PBX, Asterisk es capaz de interconectar un número determinado de teléfonos, para hacer llamadas entre sí, y conectarse a la red telefónica conmutada (PSTN) para poder realizar y recibir llamadas de usuarios conectados a esta red.

Asterisk fue creado por Mark Spencer en 1999, luego de la aparición del Proyecto Telefónico Zapata, concebido por Jim Dixon. El Proyecto Telefónico Zapata se inspiró en los increíbles avances en las capacidades y disminución de los costos de los microprocesadores modernos para plantear un sistema telefónico más económico que realizara el procesamiento de la voz por software. De esta manera se crearon los primeros diseños de tarjetas telefónicas que sirven de interfaz entre la

computadora y los circuitos telefónicos tradicionales. Con la llegada de Asterisk, se diseñaron los controladores de estas tarjetas para el sistema operativo GNU/Linux y se mejoraron las características de las mismas, al mismo tiempo que se desarrollaron múltiples aplicaciones de telefonía.

En la actualidad existen tarjetas de interface PCI para conectar tanto troncales analógicas como troncales digitales (BRI y PRI). Existen modelos para conectar hasta 24 troncales analógicas y modelos para conectar hasta 4 enlaces PRI. En la figura 7 se muestra uno de los modelos más comunes, de 4 puertos analógicos.

Asterisk incluye muchas facilidades que anteriormente solo estaban disponibles en costosos sistemas PBX propietarios, como buzón de voz, conferencias, contestadora automática y colas de llamadas, entre otras. Además es posible programar nuevas facilidades en cualquier lenguaje soportado por Linux, por ejemplo Perl, PHP y C.



Figura 7. Tarjeta de comunicaciones para conectar hasta cuatro líneas analógicas

Fuente: Guadamuz,E. (2007).

Arquitectura Asterisk (Méndez, F. (2009)): El modelo que se siguió a la hora de crear la arquitectura de *Asterisk* permite que la aplicación sea altamente escalable

y flexible. La arquitectura de Asterisk es modular, lo que permite construir un sistema funcional en base a requerimientos particulares según el caso.

En el centro de la *figura 8, Arquitectura de Asterisk*, se encuentra el núcleo de la arquitectura llamado *PBX Core*, el cual provee al resto del sistema de la funcionalidad básica y posibilita la carga de los distintos módulos, que es lo que dota a *Asterisk* de su gran funcionalidad.

Un módulo importante es el de aplicaciones y funciones. En él se encuentran las utilidades que más se utilizan a la hora de configurar *Asterisk*. Por ejemplo, para dotar a *Asterisk* de servicio de buzón de voz se hace uso de la utilidad *VoiceMail*, para que *Asterisk* pueda crear conferencias se hace uso de la utilidad *MeetMe*, para música en espera *MusicOnHold*. Existen multitud de utilidades que adicionalmente se pueden personalizar.

Otros módulos, como *formatos*, permite a *Asterisk* la gestión de los archivos *.gsm*, *.wav* utilizados por lo general para las utilidades de música en espera, buzón de voz, entre otros. El módulo de *codecs* es el que permite a *Asterisk* codificar/decodificar la información tanto de audio como de video.

En definitiva, mediante esta arquitectura se obtiene un sistema muy escalable y flexible. Escalable porque el funcionamiento independiente de los módulos permite que a la hora de crear uno nuevo, solo sea necesario tener en cuenta como se van a comunicar con el resto y flexible porque la configuración del sistema se puede hacer más ó menos compleja, a la vez que funcional dependiendo de las necesidades que deban cubrirse.

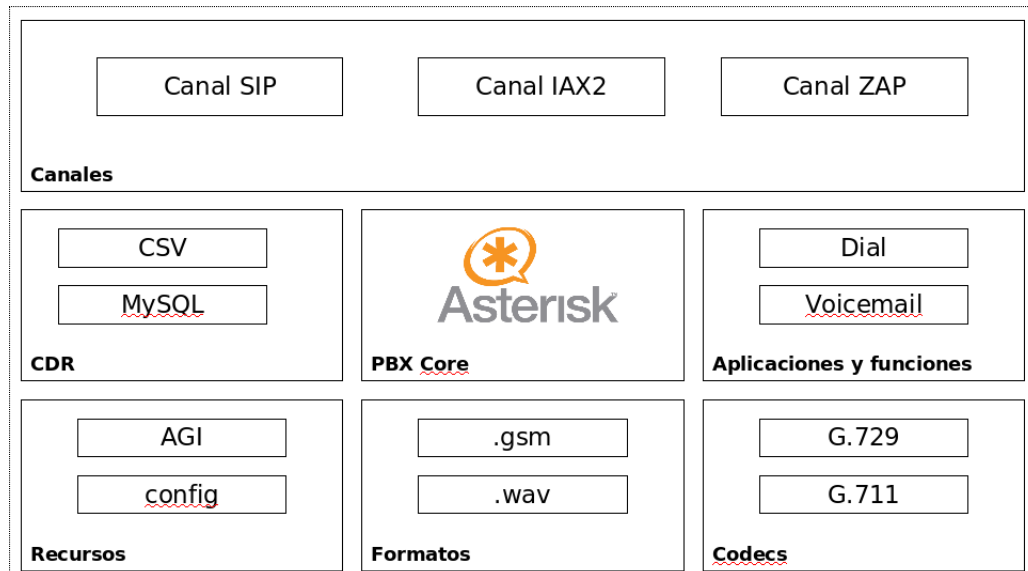


Figura 8. La Arquitectura de Asterisk

Fuente: Méndez, F. (2009).

Consideraciones para el dimensionamiento del hardware: Tomando como referencia las consideraciones estipuladas según Guadamuz, E. (2007), en términos generales la capacidad del procesador y la cantidad de memoria RAM que debe tener el servidor donde se va a instalar Asterisk puede escogerse de acuerdo a la cantidad de llamadas simultáneas que se esperan en el sistema, como se indica en el cuadro 3.

Cuadro 3.

Guía de requisitos del sistema

<i>Propósito</i>	<i>Llamadas simultáneas</i>	<i>Características mínimas recomendadas</i>
Sistemas doméstico/ microempresa	Menos de 5	600 Mhz con 128 MB RAM
Oficina Pequeña	de 5 a 10	1 Ghz con 256 MB RAM
Empresa Mediana	de 10 a 24	3 Ghz con 512 MB RAM
Sistemas medianos a grandes	Mas de 30	Procesadores duales / arquitectura distribuida

Fuente Guadamuz, E. (2007)

Hay que tomar en cuenta que la carga impuesta por el tipo de *códec* que se esté utilizando es un factor crítico, así como la necesidad o no de hacer traducciones entre *códecs* diferentes. Por ejemplo, un sistema que funciona perfectamente utilizando el *códec* G.711 podría soportar solamente el 20% de la capacidad si se utiliza el *códec* G.729.

La lógica del plan de marcación impone también cargas al procesador, especialmente cuando se hacen llamadas a aplicaciones externas para que tomen el control de la llamada. Aplicaciones como conferencias, grabación de llamadas y colas de llamadas pueden absorber gran parte de la capacidad del CPU y conllevar a un servicio deficiente.

Las tarjetas de comunicaciones de interfaz PCI son altamente sensibles a la latencia por interrupciones al procesador, por lo que son intolerantes al uso de IRQ compartidas. Se recomienda que la tarjeta madre del servidor de Asterisk permita configurar las IRQ de manera que se asigne una exclusivamente para las tarjetas de comunicaciones.

Algunos modelos de las tarjetas de comunicaciones incluyen un módulo para la cancelación de eco por hardware, lo cual puede ser recomendable para un óptimo rendimiento del sistema cuando se utilizan muchas troncales.

Instalación y configuración de Asterisk: muchos expertos, como por ejemplo Gomillon, D. y Dempster, B. (2006), aseguran que: “Aunque existen algunas versiones de Asterisk diseñadas para que se ejecuten sobre otros sistemas operativos, está mejor soportado y es más estable cuando se instala sobre el sistema operativo Linux“. Es posible instalar Asterisk en cualquier distribución de Linux, mientras ejecute el kernel 2.6.x. En la práctica, la mayor parte de las instalaciones están hechas sobre distribuciones basadas en Red-Hat, incluyendo CentOS y Fedora.

La configuración de Asterisk se realiza mediante archivos de configuración para controlar los diferentes aspectos del sistema, tales como parámetros de los protocolos, declaración de las extensiones, parámetros de las troncales analógicas y

digitales, los archivos de audio para música en espera y registros de duración de las llamadas y errores del sistema.

El plan de marcación de Asterisk es todo un lenguaje de programación, que incluye: declaración de variables globales y locales, estructuras de control, funciones, declaración de macros, paso de argumentos, patrones de marcación, entre otras cosas. Incluso Asterisk tiene la capacidad de obtener una configuración externa mediante la consulta a una base de datos, sistema conocido como Arquitectura de Tiempo Real Asterisk (ARA, del inglés *Asterisk Realtime Architecture*).

El protocolo IP junto con los protocolos de transporte TCP y UDP se encargan de llevar los paquetes desde su origen hasta el destino. Sin embargo, para manejar los procesos relativos al establecimiento, control y finalización de las llamadas se requiere un protocolo de alto nivel. Los protocolos más comúnmente utilizados en sistemas de Telefonía IP basados en Asterisk son el protocolo SIP (del Inglés *Session Initialization Protocol*), que también es el más utilizado entre fabricantes de equipos de Telefonía IP actualmente, y el protocolo IAX (del Inglés *Inter Asterisk Exchange*), el cual es ampliamente usado para la interconexión de servidores Asterisk.

SIP soporta sesiones unidifusión y multidifusión, así como llamadas punto a punto y multipunto. También soporta una variedad de servicios, como localizar al invocado y determinar las capacidades de éste, así como manejar los mecanismos de establecimiento y control de la llamada.

También es ampliamente utilizado el protocolo DUNDi (del inglés *Distributed Universal Number Discovery*), el cual no es un protocolo de señalización de VoIP como SIP, sino un estándar para la localización de puertas de enlace a servicios telefónicos. El protocolo DUNDi permite a un servidor Asterisk transmitir a otros nodos conectados directamente a él las peticiones de números telefónicos que hacen los usuarios y que el servidor no puede resolver por sí mismo. Los nodos pueden propagar la petición a otros servidores hasta encontrar el destinatario deseado.

Sistemas de Alta Disponibilidad

Colobran, M. y Otros (2008), señalan que, “Alta Disponibilidad es la capacidad de mantener operativas las aplicaciones eliminando las paradas del sistema de información”. Al respecto , estos autores también clasifican las paradas que pueden ocurrir en un sistema como: *Paradas Planificadas*, las cuales son eventos planificados para realizar mantenimientos a nivel de hardware o se software y deben coordinarse para garantizar la continuidad en el funcionamiento de la aplicación o sistema; y *Paradas no Planificadas*, las cuales son causadas por algún mal funcionamiento ó falla a nivel de hardware ó software, o por algún desastre ya sea de tipo natural (inundaciones, terremotos, incendios) o de tipo no natural (error humano ó sabotaje). En este tipo de paradas radica la importancia de diseñar sistemas basados en alta disponibilidad, que permitan la rápida recuperación de los servicios prestados al usuario en el menor tiempo posible y con el menos impacto.

Según Martínez, V. y Fuentes, A. (2009): “Un Sistema de Alta Disponibilidad consiste en que el servicio que se está dando, está lo suficientemente redundado como para que nunca pueda verse interrumpido, ni siquiera, en caso de incidencia” (Pág. 229). La disponibilidad es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su uso por el usuario final a lo largo de un tiempo dado. Ésta no sólo se relaciona con la prevención de caídas del sistema (también llamadas tiempos fuera de línea), sino incluso con la percepción de "caída" desde el punto de vista del usuario: cualquier circunstancia que le impida trabajar productivamente con el sistema. Desde tiempos de respuesta prolongados, escasa asistencia técnica o falta de estaciones de trabajo disponibles es considerada como un factor de baja disponibilidad.

Marco de análisis para determinar los requisitos de Alta Disponibilidad

Un análisis de los requisitos empresariales de alta disponibilidad y la comprensión de los costes que van incluidos, permite una solución óptima que satisfaga las necesidades de los directores de empresa, dentro de las posibilidades financieras que estarán dispuestos a pagar. A continuación se plantea un marco simple que se puede utilizar con eficacia para evaluar los requisitos de alta disponibilidad que una empresa necesita., según Aragón y Girona (2011).

Análisis del impacto de negocios: Un riguroso análisis de impacto identifica los procesos críticos de negocio dentro de una organización, calcula pérdida cuantificable sobre previstas e imprevistas caídas del sistema que afectan a cada uno de estos procesos de negocio, y se esbozan los efectos menos tangibles de estas caídas. Tiene en cuenta las funciones esenciales de la empresa, personas y recursos del sistema, las regulaciones gubernamentales y las dependencias empresariales internas y externas dependencias. Este análisis se hace usando datos objetivos y subjetivos recogidos de entrevistas con expertos y personal experimentado, revisando historiales de prácticas de empresas, informes financieros, los sistemas de registros, y así sucesivamente. Como parte de esta análisis se deben considerar tres factores importantes: **(a) Fiabilidad**, los componentes hardware fiables de una solución de Alta Disponibilidad, el software fiable, incluida la base de datos, servidores Web y aplicaciones, es la parte crítica de una implementación de una solución de alta disponibilidad; **(b) Recuperación**, puede haber muchas opciones para recuperarse de un fracaso si ocurre alguno. Es importante determinar qué tipo de fallos pueden ocurrir en su entorno de alta disponibilidad y la forma de recuperarse de estos fallos en el tiempo que satisface las necesidades de la organización; **(c) Detección de errores**, si un componente en su arquitectura falla, entonces la rápida detección, de dicho componente es esencial en la recuperación de un posible fracaso inesperado.

En el análisis del impacto sobre las empresas se clasifican los procesos de negocio, basados en la importancia del impacto sufrido con las paradas del sistema relacionadas con las IT(IT, del inglés Information Technology). Esto conlleva al siguiente elemento en el marco de los requerimientos de alta disponibilidad: el coste por tiempo de inactividad.

Coste de inactividad: Un buen análisis de impacto en las empresas proporciona información detallada sobre los costes que se derivan de tiempos de parada no planificados y planificados de los sistemas de apoyo a los diversos procesos de negocio. La comprensión de este coste es esencial porque esta tiene una influencia directa sobre la disponibilidad de alta tecnología que se utiliza para reducir al mínimo el riesgo de inactividad.

Tiempo de recuperación Objetivo (RTO, del inglés Recovery Time Objective): Se define como la cantidad máxima de tiempo que un negocio basado en procesos de IT puede estar caído antes de que comience a sufrir importantes pérdidas materiales. RTO indica la tolerancia de inactividad de un proceso del negocio o de una organización en general.

El RTO variará proporcionalmente a la naturaleza de tarea crítica del negocio en cuestión, por ejemplo en un sistema que esté ejecutando una bolsa de valores, el RTO es cero o muy cercano a cero. Una organización probablemente puede tener diferentes RTO según las necesidades de sus diversos procesos.

Punto de Recuperación Objetivo (RPO, del inglés Recovery Point Objective): Se define como la cantidad máxima de datos basados en procesos de negocio de IT, que pueden ser perdidos antes de causar un daño perjudicial para la organización. RPO indica la tolerancia de pérdida de datos de un proceso de negocio o una organización en general. Esta pérdida de datos a menudo se mide en términos de tiempo, por ejemplo, 5 horas o 2 días de valor de pérdida de datos.

Métodos para garantizar Alta Disponibilidad

Acuerdo de Nivel de Servicio

Vera, E. (2008), señala lo siguiente:

Todo sistema debe tener establecido un Acuerdo de Nivel de Servicio (Service Level Agreement - SLA) que defina cuánto tiempo y en qué horarios debe estar en línea. En el caso de aplicaciones de baja criticidad, dicho SLA puede ser de 8x5 horas a la semana excluyendo días festivos; para sistemas con mayor criticidad como una red de cajeros automáticos se tienen niveles de servicio que alcanzan las 24 horas al día, los 365 días del año.

Así entonces, suponiendo un sistema con un SLA de 24x365 se puede calcular su disponibilidad de la siguiente manera:

$$\text{Disponibilidad} = ((A - B)/A) \times 100 \text{ por ciento}$$

Donde:

A = Horas comprometidas de disponibilidad: $24 \times 365 = 8,760$ Horas/año.

B = Número de horas fuera de línea (Horas de "caída del sistema" durante el tiempo de disponibilidad comprometido). Por ejemplo: 15 horas por falla en un disco; 9 horas por mantenimiento preventivo no planeado.

De esta manera:

$$\text{Disponibilidad} = ((8,760 - 24)/8,760) \times 100 \text{ por ciento} = \mathbf{99.726\%}$$

Cuando se realicen negociaciones para definir objetivos de disponibilidad con los usuarios, es necesario hacerlos concientes de las implicaciones técnicas y económicas, como se muestra en el cuadro 4

Cuadro 4.

Disponibilidad de un sistema 24x7 y tiempos de caída permitidos

Disponibilidad (%)	Tiempo offline/año	Tiempo offline/mes	Tiempo offline/día
90%	36.5 días	73 hrs	2.4 hrs
95%	18.3 días	36.5 hrs	1.2 hrs
98%	7.3 días	14.6 hrs	28.8 min
99%	3.7 días	7.3 hrs	14.4 min
99.5%	1.8 días	3.66 hrs	7.22 min
99.9%	8.8 hrs	43.8 min	1.46 min
99.95%	4.4 hrs	21.9 min	43.8 s
99.99%	52.6 min	4.4 min	8.6 s
99.999%	5.26 min	26.3 s	0.86 s
99.9999%	31.5 s	2.62 s	0.08 s

Fuente: Viera, E. (2008).

Estos números (especialmente aquellos que pasan de la marca del 99.5% de disponibilidad) son difíciles de alcanzar, ya que es necesario poder recuperarse ante caídas del sistema de manera transparente. La capacidad e intervalo de tiempo necesarios para recuperarse ante tal eventualidad son directamente dependientes de: (a) La complejidad del sistema; (b) La severidad del problema; (c) La disponibilidad del personal de soporte; (d) La madurez en materia de administración del sistema y sus operaciones; (e) Otros factores técnicos o de gestión (falta de refacciones, fallas en la cadena de escalamiento, entre otros).

Administración de Sistemas

La administración de sistemas es parte integral de cualquier sistema con alta disponibilidad al elaborar un plan eficaz para hacer frente a los retos que disminuyen la disponibilidad de un sistema. Antes de elaborar dicho Plan, es necesario primero comprender el sistema en su conjunto y cómo cada componente afecta la disponibilidad general del mismo. No importa lo insignificante que pueda parecer un

componente del sistema, es posible que tenga un profundo impacto sobre la disponibilidad general del sistema.

Una vez que hayan sido identificados los componentes más críticos, se puede buscar la manera de mejorar su confiabilidad, capacidad de servicio, recuperación y administración.

Redundancia

Según Viera, E. (2008), “La redundancia es una técnica mediante la cual un componente del sistema es duplicado y cualquiera de sus instancias puede ser utilizada en caso de falla”. Entre los componentes redundantes debe existir una configuración que permita el re-balanceo de la carga de manera automática en caso de que alguno de los componentes dejara de funcionar y así, el sistema puede continuar su funcionamiento y no debe existir impacto alguno en la operación si es que esto llegara a ocurrir.

Algunas de las cuestiones que deben tomarse en cuenta para que un sistema sea tolerante a fallos son: (a) Redundancia en el suministro eléctrico; (b) Redundancia de discos duros; (c) Redundancia de las conexiones de red.

Una de las principales desventajas de este método radica en los altos costos en los que se incurre; sin embargo, actualmente ha vuelto a cobrar fuerza una técnica que permite emular la redundancia del hardware por medio de software especializado (Vmware, Xen, VirtualBox, entre otros). Esta técnica se conoce como virtualización.

Configuración de Cluster: Los sistemas basados en clusters se implementan con el objetivo de mejorar la disponibilidad de los servicios que se ofrecen. Este método se puede ver como un complemento del método de redundancia en el caso de la prestación de servicios, ya que permite sincronizar los componentes redundantes de un sistema para que sean vistos por el usuario final como uno solo y estén disponibles en cualquier momento.

No se debe confundir clúster de alta disponibilidad con clúster de alto rendimiento. El primer tipo de clúster está diseñado para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones, mientras que el segundo es una configuración de equipos diseñado para proporcionar capacidades de cálculo mucho mayores que la que proporcionan los equipos individuales.

Colobran, M. y Otros (2008), enuncian lo siguiente: “Los cluster de alta disponibilidad están destinados a proporcionar disponibilidad de recursos y servicio mediante redundancia”, mientras que Martín, V. y Fuentes, A. (2009), señalan:

El diseño y la aplicación de un cluster de alta disponibilidad es una tarea compleja que requiere tanto de software específico como de una arquitectura hardware singular. La configuración del software del cluster no es sencilla ya que se deben tener en cuenta muchos factores. En primer lugar el servidor considerado como nodo maestro y el correspondiente al nodo o nodos esclavos deben tener los mismos sistemas operativos y programas instalados. Además tienen que poder acceder a los mismos ficheros de datos, que deben ser actualizados en tiempo real para que no se produzcan errores de concordancia y asegurar la conmutación automática entre ambos nodos en caso de que se produzca un fallo en el hardware de alguno de ellos. (Pág. 29).

Los clusters de alta disponibilidad normalmente están conectados por unas redes privadas que permiten monitorizar el estado de todos los nodos del cluster y disponer de un entorno de almacenamiento distribuido compartido. Cada nodo envía continuamente unas señales que reciben el nombre de “heartbeats” a los demás nodos para indicarles cuál es su estado actual.

Por otro lado, Alsina, E. (2009), hace referencia a que la tecnología de clúster puede ser implementada de dos formas básicas: **(a) Clúster activo-activo**, donde todos los equipos que conforman el clúster prestan el mismo servicio a la vez. Es también llamado clúster de balanceo de carga. **(b) Clúster activo-pasivo**, en el cual, si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del clúster, los servicios que han fallado arrancan automáticamente en cualquiera de

las otras máquinas del clúster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original

Virtualización: Permite construir infraestructuras de máquinas y redes virtuales que emulan a máquinas y redes físicas convencionales proporcionando gran transferencia en funcionalidad y rendimiento.

Una definición más formal de virtualización es la expuesta por Lillo, S. (2007), durante el desayuno tecnológico organizado por la empresa Coriolis ITS en la ciudad de Santiago de Chile, “Es una capa de hw/sw que esconde los elementos reales para presentar unos recursos e información (sistemas virtuales) al usuario y/o gestor sin necesidad de acceder directamente a los elementos discretos.”.

El mismo autor señala tres propiedades de la virtualización: **(a) Particionamiento**, donde existen múltiples sistemas operativos en el mismo servidor, utilizando por completo los recursos del mismo. Esta propiedad se ilustra en la figura 9; **(b) Aislamiento**, ya que la virtualización asigna espacios independientes en CPU, RAM y disco a cada sistema operativo y controla la asignación de recursos para cada máquina virtual. Esto aísla posibles fallos en un determinado sistema invitado tal y como se muestra en la figura 10, donde un virus ataca a uno de los sistemas invitados pero el resto sigue funcionando sin problemas; **(c) Encapsulación**, puesto que la virtualización crea máquinas virtuales las cuales son gestionadas como archivos (Ver figura 11), lo que las hace además completamente portátiles.

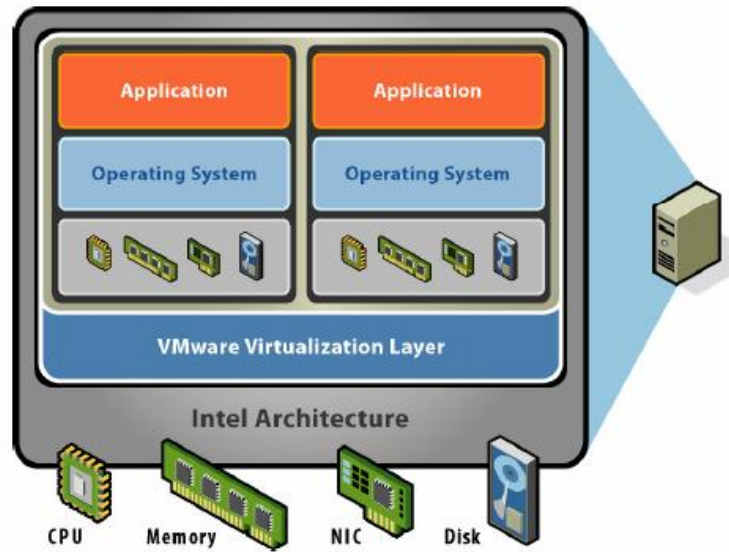


Figura 9. Propiedades de la virtualización: Particionamiento

Fuente: Lillo, E. (2007).



Figura 10. Propiedades de la virtualización: Aislamiento

Fuente: Lillo, E. (2007).

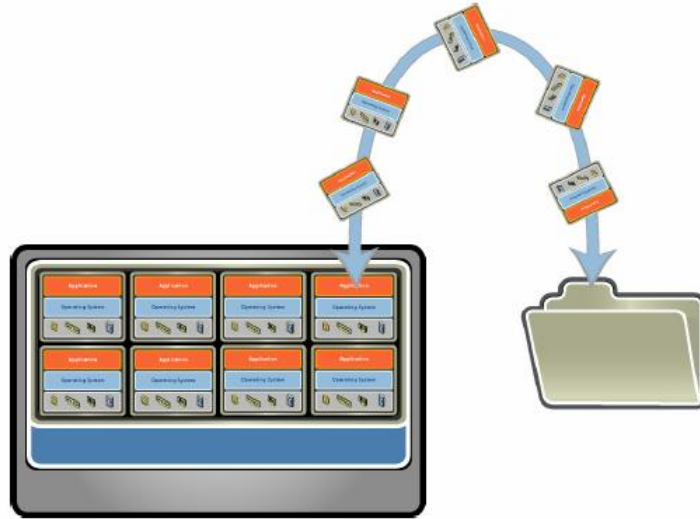


Figura 11. Propiedades de la virtualización: Encapsulación.

Fuente: Lillo, E. (2007).

Dicho de otra manera, la virtualización describe como en un sólo computador físico se coordina el uso de los recursos para que varios sistemas operativos puedan funcionar al mismo tiempo de forma independiente y sin que estos sepan que están compartiendo recursos con otros sistemas operativos.

Respaldo de recursos críticos

Es una técnica donde un componente del sistema (generalmente de software o la información de la base de datos) es duplicado y el respaldo es puesto en reserva, para ser usado en caso de que el componente primario falle. A diferencia de la redundancia, el usuario final puede percibir una caída del sistema; sin embargo, el respaldo se emplea para minimizar el tiempo que toma hacer el cambio hacia el componente de respaldo, reduciendo a su vez el tiempo que transcurra el sistema fuera de línea.

Aunque el respaldo de elementos de software tales como la información de base de datos, bitácoras o configuración del sistema operativo es el más común, también existen otros dos tipos de respaldo:

1. **De componentes de hardware:** dependiendo del diseño del sistema, los elementos respaldados son generalmente aquellos con partes mecánicas, manufactura de menor calidad o que están sujetos a condiciones ambientales extremas (por ejemplo, en algunos países y regiones el suministro de energía eléctrica no es confiable con apagones y altibajos de voltaje frecuentes; en dichas localidades las fuentes de poder o reguladores de energía tienen un tiempo de vida mucho menor).
2. **Del personal operativo de Tecnología:** Es indispensable contar siempre con más de una persona con los conocimientos críticos necesarios para asegurar que la operación del sistema se lleve a cabo a pesar de la ausencia o incluso pérdida de personal clave. Esto se logra mediante manuales de procedimientos claramente definidos y contratos de mantenimiento sobre los componentes del sistema.

En muchas compañías (especialmente aquellas que poseen infraestructura de alto costo o con operaciones que involucren información crítica) es indispensable que los respaldos de hardware, software y personal clave así como los procedimientos de restauración de los mismos estén integrados en un Plan de Recuperación ante Desastres.

Balanceo de tráfico

Según lo señalado por Viera, E. (2008), este método consiste en un equipo que distribuye peticiones o flujos de red entre un conjunto de servidores conectados a la misma, para equilibrar la carga de trabajo entre ellos. De esta manera, se optimiza el uso de los recursos, su rendimiento o el tiempo de respuesta global del sistema. La técnica de balanceo de tráfico aporta fiabilidad al sistema gracias a la redundancia

que consigue. Adicionalmente, estos sistemas de balanceo de tráfico pueden adaptar sus decisiones de balanceo de tráfico en función de las modificaciones de la carga del sistema, lo que se conoce como balanceo de carga, o incluso del tipo de peticiones que se realicen.

A la hora de diseñar un servicio de balanceo de tráfico, es muy importante seleccionar una política de balanceo adecuada para determinar a qué servidor se envía cada una de las peticiones de servicio recibidas. Estas políticas pueden clasificarse en dos grandes grupos: **(a) No adaptativa**, que utiliza algoritmos que no se ajustan al estado actual del sistema, sino que realizan la elección del servidor de manera aleatoria; **(b) Adaptativa**, la cual emplea algoritmos que utilizan información en tiempo real que les permite elegir el servidor más adecuado para cada petición en concreto.

Estandarización

Nuevamente, tomando como referencia lo expuesto por Viera, E. (2008) con respecto a la estandarización, se puede definir como el establecimiento de especificaciones para el hardware, software, procedimientos y técnicas que se utilizarán en toda la infraestructura de sistemas. Esta disciplina es a menudo ignorada en favor de dar a los usuarios la libertad de elegir sus propios conjuntos de aplicaciones e interfaces con los demás sistemas de la empresa. Sin embargo, la libertad sin restricciones a menudo conduce al caos y problemas con la administración del sistema. Por otro lado, cualquier modelo de gestión de calidad (como CMM, Six Sigma o ISO-9000) requiere del establecimiento de un sistema de normas y estándares corporativos.

Por lo general, un sistema de Alta disponibilidad se va diseñar integrando mas de uno de los métodos anteriormente explicados los cuales entran en funcionamiento gracias a elementos de software analizados a continuación.

Software de Alta Disponibilidad

Las siguientes soluciones de alta disponibilidad basadas en software, fueron documentadas por Méndez, F. (2009) y están destinadas a ofrecer alta disponibilidad en servicios y alta disponibilidad en datos.

Alta Disponibilidad en Servicios

Las soluciones de alta disponibilidad enfocada a los servicios son especialmente utilizadas en servicios críticos. Normalmente, estos servicios irán acompañados de complementos que se requieren para su funcionamiento tales como: direccionamiento IP, reglas de firewall, copias de seguridad, etc. Por ejemplo, la mayoría de las empresas requieren de la utilización del servicio de Directorio Activo por lo que este se convierte en este caso en un servicio crítico. Una configuración de alta disponibilidad podría consistir por ejemplo en dos nodos donde solo uno de ellos dispone de una dirección IP Virtual, mediante la cual las aplicaciones acceden al Directorio Activo.

En caso de fallo del nodo que ofrece el servicio, la dirección IP Virtual se establece en el otro nodo (nodo secundario o backup) lo que permite a los clientes y aplicaciones seguir accediendo al Directorio Activo como si no hubiera ocurrido nada. Algunas de las herramientas GNU/Linux enfocadas a la alta disponibilidad en servicios son: KeepAlived, Heartbeat, entre otros.

Alta Disponibilidad en Datos

Las soluciones de alta disponibilidad enfocadas a los datos son fundamentales y, por lo general, es necesaria su integración con los sistemas de alta disponibilidad enfocados a servicios. El objetivo que se persigue con la alta disponibilidad enfocada a los datos es mantener en varias localizaciones una serie de datos, los cuales no

presentan problemas de consistencia ni integridad entre ellos. En GNU/Linux se dispone de varias herramientas software que cumplen este cometido. Estas herramientas se pueden clasificar en dos categorías:

1. **Replicación de bloque de datos:** La replicación de bloque de datos consiste en replicar la información bloque a bloque del sistema de ficheros. Una de las herramientas que se utilizan para la replicación de datos es DRBD o Distributed Replicated Block Device, un software de replicación de dispositivos de bloque (discos duros, particiones, volúmenes) creando discos duros espejo en distintos servidores. Algunas de sus características son una replicación en tiempo real de manera continua, transparencia en las aplicaciones que estén almacenando datos en la unidad ya que no sabrán que está se está replicando en varias localizaciones, etc. La principal limitación que presenta DRBD es cuando los cambios se están realizando en más de una localización a la vez, ya que la “unión” de los datos de más de una localización no es posible ya que solo se permite tomar una de las localizaciones como modelo de copia, siendo el resto esclavos. Otra herramienta de este tipo es rsync, una aplicación para sistemas de tipo Unix que ofrece transmisión eficiente de datos incrementales comprimidos y cifrados.
2. **Replicación de bases de datos:** La replicación de bases de datos consiste en replicar una base de datos en diversas bases de datos remotas. A diferencia de lo anterior, la replicación se lleva a cabo sentencia a sentencia. Gracias a la flexibilidad de diseño que permiten las bases de datos actualmente es posible realizar modificaciones en las bases de datos de más de una localización. Flexibilidad presente en funcionalidades como la Replication de MySQL y a sus atributos como `auto_increment_increment` y `auto_increment_offset` permiten superar la limitación que tienen las soluciones basadas en bloque de datos. En GNU/Linux se dispone también

de otras muchas herramientas de replicación de base de datos como Slony-I, además de la replicación que ofrece Oracle entre muchos otros.

Proyectos de Alta Disponibilidad

Existe una gran variedad de proyectos que están desarrollando actualmente el tema de la alta disponibilidad. Dentro del trabajo realizado por Méndez, F.(2009), se desglosan proyectos tanto para la alta disponibilidad en servicios como en datos.

Proyectos destinados a ofrecer Alta Disponibilidad en servicios

1. **HA-OSCAR:** Es un proyecto cuyo objetivo es proporcionar de manera combinada el poder de la alta disponibilidad con un gran rendimiento de cómputo. *HAOSCAR* tiene en cuenta los puntos débiles por lo que adopta redundancia de componentes e incorpora detección de fallos, mecanismos de recuperación, entre otros.
2. **The High Availability Linux Project (Heartbeat):** Es una de las soluciones más ampliamente extendidas. Se integra perfectamente con multitud de servicios y sigue aún mejorando dicha integración gracias a la aportación de su activa comunidad. *Heartbeat* es un *software* que ofrece alta disponibilidad a determinados recursos mediante la creación y mantenimiento de un clúster compuesto por una serie de nodos. Los recursos se ejecutan y se mueven entre los distintos nodos ya sea por motivos de fallo o simplemente por motivos de administración.
3. **Kimberlite:** Es considerado un proyecto único ya que es una completa infraestructura de alta disponibilidad de código abierto que incluye incluso garantía sobre la integridad de los datos. Actualmente es un proyecto abandonado o de muy baja actividad.

4. **LifeKeeper.** Es una alternativa moderna y flexible comparada con los productos que ofrecen soluciones de alta disponibilidad. Dispone de mecanismos para mantener la integridad de los datos. Una numerosa colección de kits de recuperación para los distintos servicios y/o aplicaciones permiten instalar *LifeKeeper* en cuestión de horas. Es un producto de la empresa *SteelEye*.
5. **HP Serviceguard:** Es un software especializado para ofrecer alta disponibilidad a las aplicaciones más críticas ante fallos tanto hardware como software. *HP Serviceguard* monitoriza el estado de cada nodo y responde rápidamente en caso de fallo permitiendo minimizar el tiempo durante el cual no se está ofreciendo el servicio. Está disponible tanto para Linux como para HP-UX.
6. **Red Hat Cluster Suite:** Ha sido diseñado específicamente para Red Hat Enterprise Linux. Proporciona además de alta disponibilidad balanceo de carga. Dispone de soluciones listas para implantar para la mayoría de los servicios, pudiendo además proporcionar soporte cuando sea necesario.

Proyectos destinados a ofrecer Alta Disponibilidad en datos

1. **DRBD:** *Software* de replicación de dispositivos de bloque formando un *RAID* 1 a través de la red, es decir, replica los datos en distintas localizaciones. Algunas de sus características son: una replicación en tiempo real de manera continua, transparencia en las aplicaciones que estén almacenando datos en la unidad, posibilidad de recuperación de los datos ante un desastre, entre otras. Se complementa perfectamente con el proyecto de alta disponibilidad en servicios *Heartbeat*.
2. **FreNAS:** es un servidor *NAS* “*Network-Attached Storage*” gratuito, que soporta: protocolos *Cifs* (*Samba*), *Ftp*, *Nfs*, *Rsync*, autenticación de usuario local, *RAID* (0,1,5) por *software* con un completo interfaz de

configuración Web. *FreeNAS* ocupa menos de 32MB una vez instalado en *Compact Flash*, disco duro o un dispositivo de memoria *usb*. Está basado en *FreeBSD* que es un derivado de *Unix*, permitiendo clientes tanto de *GNU/Linux* como *Windows* y por supuesto *Unix*.

3. **Openfiler.** Provee de buena integración con *iSCSI* (Internet *SCSI*) y “*fibre channel*”, muy útiles en entornos empresariales y en virtualización tal como *Xen* y *VMware*. Cuenta con una interfaz Web para su administración y soporta clientes de todos los sistemas operativos.
4. **MySQL Replication:** Es un módulo del servicio *MySQL*. Consiste en una replicación asíncrona unidireccional (un servidor actúa como maestro y uno o más actúan como esclavos).
5. **Slony-I.** Es un sistema de replicación de “maestro a múltiples esclavos”, el cuál además soporta el concepto de “cascada” (un nodo puede proveer a otro nodo, el cual puede proveer a otro) permitiendo así la recuperación ante errores. Incluye las principales características para replicar bases de datos de gran información a un número límite razonables de esclavos. Es por tanto un sistema diseñado para centros de datos y sitios de *backup* donde el funcionamiento normal requiere que todos los nodos estén disponibles en un momento determinado.

Algunos de los proyectos mencionados implican costos de licencia, otros son completamente gratuitos, otros ofrecen soporte o disponen de una gran comunidad.

Esquema General de Alta Disponibilidad (Guadamuz, E., 2007)

El principio básico de la Alta Disponibilidad es la replicación de los elementos que componen el sistema, tanto a nivel de hardware como de software. Cuando existe algún elemento dentro del sistema que no se encuentra replicado, un determinado fallo de ese elemento puede desencadenar el fallo del sistema completo.

Para conseguir altos niveles de disponibilidad en servicios de red, los sistemas deben ser tolerantes a fallas, lo que significa que deben ser capaces de seguir funcionando aún cuando se presente una falla de hardware o software. Para esto debe incluirse redundancia en los elementos que toman parte para brindar el servicio, de manera que no haya ningún punto simple de fallo en todo el sistema. Esto no solamente incluye a los servidores, sino que también el almacenamiento de la información y las rutas de red por donde viaja la información

El método básico para lograr la replicación de los elementos de un sistema es por medio de la configuración de clúster.

Los grupos de elementos que conforman el clúster, tienen que tener claramente implementados los algoritmos necesarios para que en el momento que un nodo deje de dar servicio, sea cambiado fácilmente por cualquier otro nodo del clúster, sin que el sistema deje de dar servicio, que es el objetivo fundamental. Es decir, todos los nodos del cluster tienen que estar completamente preparados para realizar su tarea en cualquier momento.

El software dedicado a la alta disponibilidad lo que debe hacer es abstraer el clúster y trabajar como si se tratase de una única entidad capaz de realizar todas estas tareas.

La forma básica de utilizar un cluster para mejorar la disponibilidad de un sistema es mediante un conjunto de dos nodos redundantes, como se muestra en la figura 12.

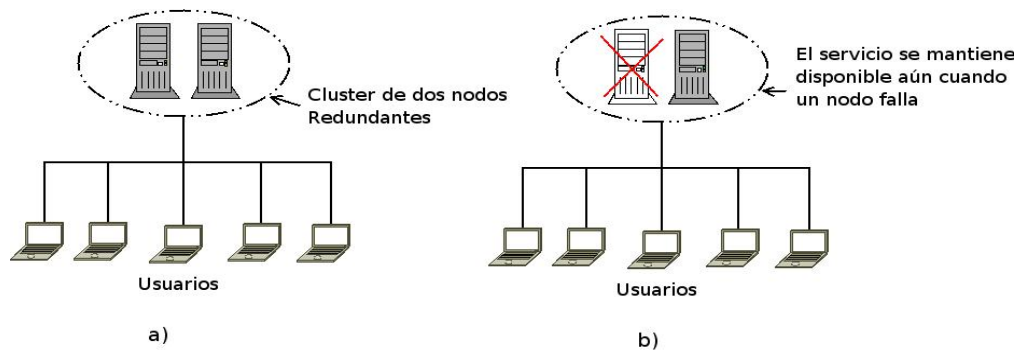


Figura 12. a) Cluster de dos nodos redundantes. b) El servicio se mantiene disponible aún cuando uno de los nodos deja de operar.

Fuente: Guadamus, E. (2007).

La alta disponibilidad, ya sea de servicios o de datos, puede ser alcanzada tanto haciendo uso de soluciones de software como de soluciones de hardware. La variedad que existe en ambos grupos es muy grande; además, es posible integrar ambos grupos para generar soluciones conjuntas. Por otra parte, a nivel de software, también se categorizan las soluciones de alta disponibilidad como propietarias y soluciones libres o gratuitas pudiendo elegir siempre una solución que mejor se adapte a las necesidades del sistema de estudio.

A continuación se estudian los esquemas más importantes orientados a ofrecer Alta Disponibilidad, específicamente a servicios de Telefonía IP.

Soluciones de Alta Disponibilidad orientadas a Telefonía IP

(Méndez, F., 2009)

El objetivo de esta sección es mostrar la posibilidad de integración de los servicios de Telefonía IP con las soluciones de Alta Disponibilidad vistas anteriormente. Existen diversas opciones a la hora de integrar alta disponibilidad en un sistema de Telefonía IP, cada una presentando sus ventajas e inconvenientes, de forma que permita adaptar las características del sistema a los requisitos que se

deseen cumplir, manteniendo los costos de implantación. Para ver el alcance de las distintas combinaciones que se pueden llevar a cabo, se plantea la arquitectura básica de un Sistema de Telefonía IP que se muestra en la figura 13, para ir complementándola e integrándola con los métodos utilizados para garantizar alta disponibilidad ya vistos en los puntos anteriores.

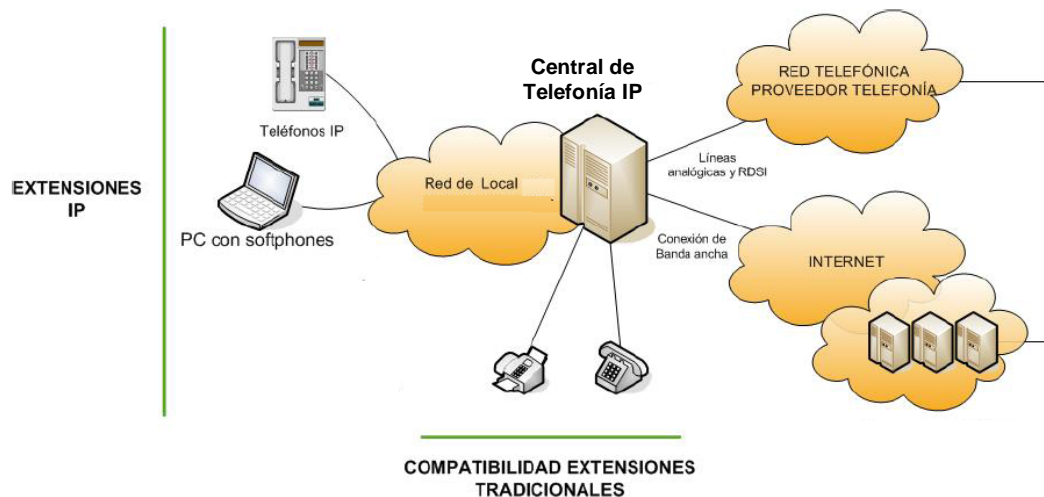


Figura 13. Arquitectura básica de un Sistema de Telefonía IP

Fuente: Méndez, F. (2009).

La arquitectura básica de un Sistema de Telefonía IP que se muestra en la figura 13 se compone de una Central de Telefonía IP sobre la cual se registran todas las extensiones de los usuarios del sistema. Esta central de Telefonía IP está conectada a dos tipos de proveedores externos: un proveedor del servicio de Internet de banda ancha para sacar el tráfico nativo de *VoIP* y otro hacia la Red Telefónica Convencional, conectado por medio de líneas analógicas y digitales, a través de las cuales se pueden realizar llamadas a la red telefónica tradicional.

Para Méndez, F. (2009) a primera vista, el sistema cuenta con dos “Puntos Simples de Fallo”. El más importante es la Central de Telefonía IP ya que, en caso que esta llegara a fallar, todos los usuarios se quedan sin el servicio, no pudiendo

realizarse ningún tipo de llamada (ni llamadas internas, ni llamadas realizadas por los usuarios externos).

El otro punto de fallo identificado por este autor, es la troncal de Internet. Si esta no se encuentra disponible durante un tiempo no se podrán realizar llamadas a través de un proveedor de *VoIP* sino que sólo se podrán realizar llamadas a través de la Red Telefónica Convencional, con el considerable incremento de costos que esto supone. Sin embargo, y afortunadamente, las llamadas entre usuarios internos sí podrían realizarse ya que internamente el servicio no se vería afectado por el fallo de la troncal.

En el caso de fallar alguna de las líneas que conectan a la Red Telefónica convencional, la situación es similar. No podrían realizarse llamadas externas a la Red Telefónica Tradicional sino sólo a proveedores de VoIP, pero de igual manera la comunicación interna entre usuarios no se vería afectada.

Con base a este escenario inicial planteado en la figura 13, se presentan a continuación los diversos métodos para garantizar Alta Disponibilidad estudiados en secciones anteriores de este capítulo, eliminando con ellos los puntos de fallo identificados:

Redundancia: Por medio de la redundancia, puede ser eliminado el principal Punto Simple de Fallo Detectado, incorporando una segunda Central de Telefonía IP la cual será una réplica exacta de la primera Central en cuanto a la configuración (posee una configuración estática idéntica). Estas deben trabajar de forma redundante y transparente al usuario final por medio de la configuración de un cluster de alta disponibilidad. Podría inclusive plantearse un escenario en el que cada central permanezca en una ubicación física distinta y de esta manera, además de solucionar el problema de fallo de *hardware/software* de uno de los dos nodos, en caso de fallo del suministro eléctrico o incendio en una de las ubicaciones físicas, el servicio de telefonía seguiría disponible.

El detalle de la configuración del cluster de alta disponibilidad dependerá de los requerimientos y características de cada sistema de Telefonía IP en particular,

pudiendo utilizar técnicas como la virtualización para complementar la redundancia y evitar incurrir en costos de implementación demasiado elevados.

Respaldo de recursos críticos: este método, puede utilizarse en conjunto con la redundancia para eliminar el principal Punto de Fallo como lo es la Central de Telefonía IP, pero más específicamente, es de gran utilidad para el respaldo de la base de datos de los usuarios, ya que es el componente que puede mantenerse duplicado con el fin de restaurarlo manualmente en caso de fallas en la base de datos original.

Balanceo del tráfico: Este método también integra la redundancia, ya que para atacar el Punto Simple de fallo del sistema de Telefonía IP, se debe contar con dos o más Centrales de Telefonía IP, para distribuir la carga de trabajo entre las mismas. Además, para garantizar que este método permita desviar el tráfico en caso de la caída de alguno de los nodos, sería recomendable utilizar una configuración de tipo Adaptativa, la cual emplea algoritmos que utilizan información en tiempo real que les permite elegir el servidor más adecuado para cada petición en concreto. Este método permite además una mayor escalabilidad al Sistema de Telefonía IP, permitiendo agregar nuevos nodos para distribuir mejor la carga del tráfico a medida que sea necesario agregar nuevas extensiones.

El esquema de Alta Disponibilidad que se diseñe para el Sistema de Telefonía de una determinada empresa o institución, dependerá del tamaño, complejidad y criticidad del servicio telefónico dentro de la misma. Se pueden plantear esquemas sencillos, los cuales utilizarían uno de los métodos anteriormente señalados ó inclusive pueden llegar a integrarse varios de estos métodos en un solo esquema.

Identificación de la Institución

Nombre: Instituto de Prevención y Asistencia Social para los empleados del Ministerio del Poder Popular para la Educación (IPASME).

Misión: Promover y desarrollar un Sistema de Seguridad Social Integral y eficaz orientado al mejoramiento permanente de la Calidad de Vida de los profesores,

maestros y personal administrativo del Ministerio del Poder Popular para la Educación y de otros organismos afiliados al Instituto de Previsión y Asistencia Social del Ministerio de Educación (IPASME), mediante la prestación oportuna, eficiente y efectiva de asistencia en sus requerimientos de afiliación, créditos: Hipotecarios o personales; médicos-asistencial; culturales, recreativos, deportivos y de ayudas socioeconómicas, siendo imprescindible para ello el establecimiento de sistemas y mecanismos que garanticen la protección social y el mejoramiento de las condiciones de vida de sus afiliados y beneficiarios.

Visión: Ser un organismo eficiente en la prestación de servicios de Seguridad y Asistencia Social Integral, con una atención de alta calidad para el personal Docente y Administrativo del Ministerio del Poder Popular para la Educación, y otros entes educativos afiliados al IPASME, en el marco de los principios de universalidad, solidaridad, unidad, integralidad y participación democrática, para convertirnos en una Institución modelo del Sistema de Seguridad Social en Venezuela y en América Latina.

Organigrama funcional

El IPASME cuenta con 64 sedes a nivel nacional, divididas en 61 unidades IPASME, 3 sedes administrativas, 2 destinos turísticos pertenecientes a la organización que no son contabilizados como sedes; Iparmar ubicado en Rio Chico, estado Miranda y el hotel Valle Grande en Mérida. A su vez las unidades IPASME están categorizadas como tipo I, II y III respectivamente de las cuales 24 son del tipo I, 23 del tipo II, 14 tipo III y se diferencian por su tamaño en estructura, la población que abarca y los servicios que presta. La figura 14 muestra las distintas unidades organizativas y su distribución dentro de la institución.

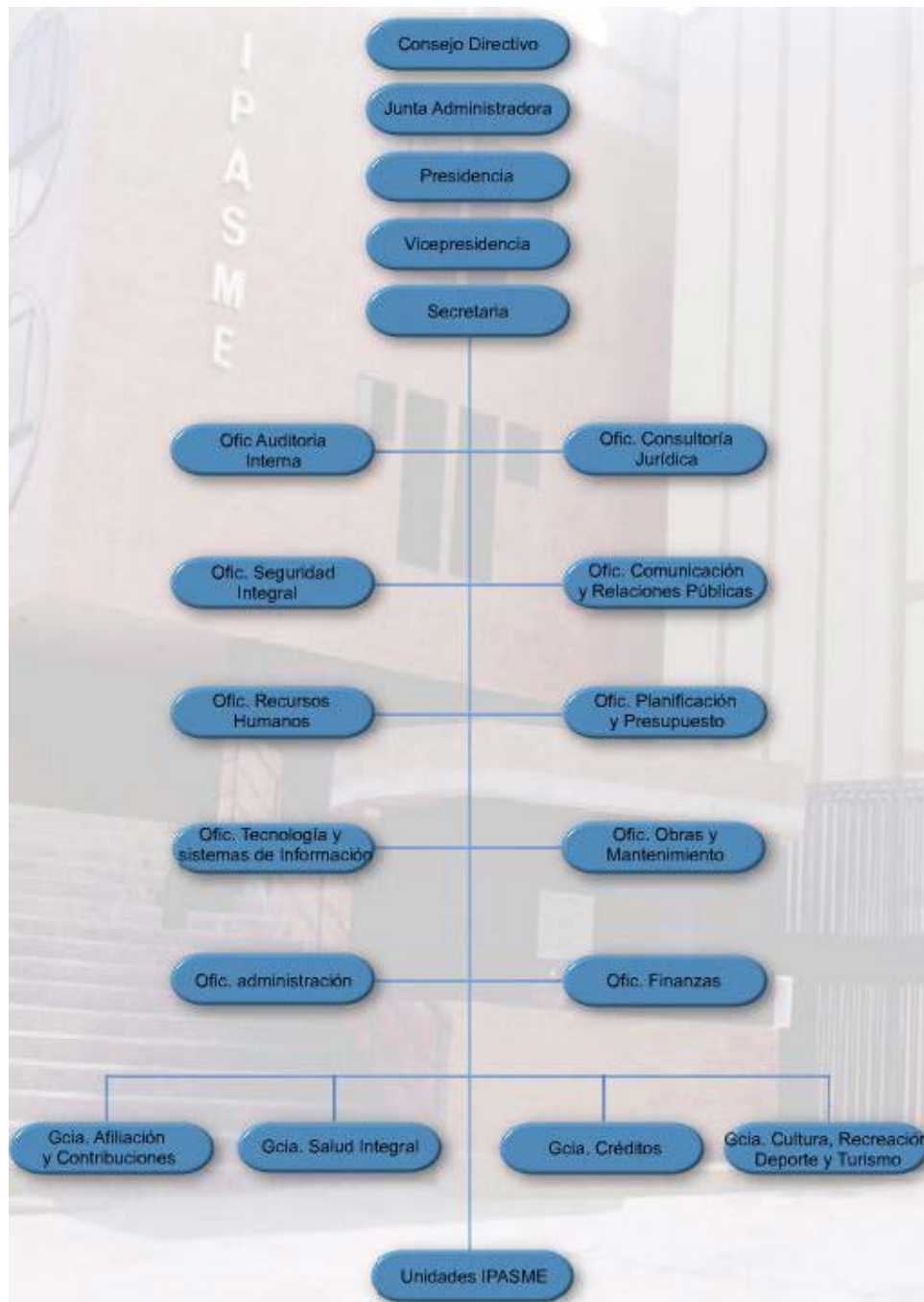


Figura 14. Organigrama funcional de la institución

Fuente: <http://www.ipasme.gov.ve>

La Oficina Central de Tecnología y Sistemas de Información (OCTSI)

Es la dependencia administrativa adscrita a la junta administradora responsable de asesorar a la alta gerencia del instituto y sus diferentes dependencias organizativas, así como proporcionar asesoría y soporte en el área de Tecnología y Sistemas de Información. La figura 15 muestra la estructura organizativa de esta oficina y donde puede identificarse la Coordinación de Redes y Comunicaciones, dependencia de OCTSI responsable directa del Sistema de Telefonía IP de la sede administrativa principal de esta institución.

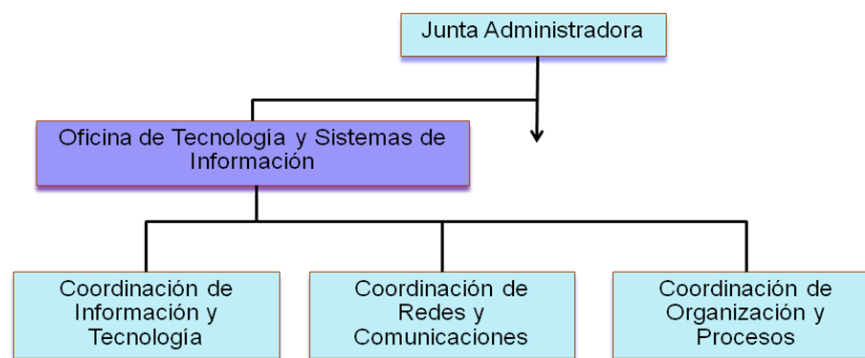


Figura 15. Estructura Organizativa de OCTSI

Fuente: Coordinación de Organización y Procesos de OCTSI (2010)

Bases Legales

Los aspectos regulatorios de la telefonía a través de la red de datos todavía no están contemplados en muchos países, debido a que sus servicios están considerados como un valor agregado de Internet y no como servicios sujetos a concesión. En el caso de Venezuela, existen aspectos contemplados dentro de la Constitución vigente y La Ley Orgánica de Telecomunicaciones que pueden tomarse como marco jurídico para la presente propuesta de investigación. Además, por tratarse de un proyecto para una institución pública, también se debe tomar en cuenta el decreto 3.390 emanado

por el ejecutivo nacional, así como los estatutos que rigen el IPASME como institución de manera de soportar este trabajo.

Constitución de la República Bolivariana de Venezuela

El marco legal en el cual se ha basado el desarrollo de las Tecnologías de Información y Comunicación en la nación venezolana, tiene su base principal en el artículo 110 de la actual Constitución de la República Bolivariana de Venezuela:

El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para las mismas. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía.

Ley Orgánica de Telecomunicaciones

La Ley Orgánica de telecomunicaciones establece dentro de sus objetivos “Promover el desarrollo y la utilización de nuevos servicios, redes y tecnologías cuando estén disponibles y el acceso a éstos, en condiciones de igualdad de personas e impulsar la integración del espacio geográfico y la cohesión económica y social”. Este planteamiento se ve reforzado dentro del cuerpo de esta Ley en su artículo 12, correspondiente a los deberes y derechos de los usuarios de los servicios de telecomunicaciones, donde se señala entre los derechos de los usuarios: “Acceder en condiciones de igualdad a todos los servicios de telecomunicaciones y a recibir un

servicio eficiente, de calidad e ininterrumpido, salvo las limitaciones derivadas de la capacidad de dichos servicios”

Este último planeamiento de la Ley Orgánica de Telecomunicaciones, constituye un marco jurídico ideal para justificar la propuesta de un Sistema de Telefonía IP de Alta disponibilidad que garantice la prestación del servicio de Telefonía IP de forma óptima e ininterrumpida para el IPASME.

Plan de Desarrollo Económico y Social de la Nación 2007-2013

El Plan de desarrollo de la nación 2007-2013, también hace una importante referencia al desarrollo e innovación tecnológica en cuanto a la prestación de servicios. Entre los objetivos del Modelo Productivo Socialista se plantea:

Se hará un énfasis particular para impulsar el logro de un desarrollo tecnológico interno que posibilite la autonomía relativa de las actividades productivas y de servicios necesarias para alcanzar y sostener el desarrollo, mediante el fortalecimiento de la capacidad de innovar, importar, modificar y divulgar tecnologías orientadas primordialmente a la satisfacción de las necesidades humanas que, a su vez, favorezcan el desarrollo científico.

Decreto Presidencial 3.390

El 28 de diciembre de 2004, se publica en Gaceta Oficial un decreto presidencial que establece que:

Artículo 1. La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollado con Estándares Abiertos.

Esto quiere decir que los paquetes de Software a usarse como soporte en las funciones de la administración pública deben ser productos de desarrolladores que practiquen la filosofía de Software libre y, por tanto, haber sido programados bajo estándares abiertos, lo cual representa un importante factor a tomar en cuenta en la selección del software utilizado para completar el diseño propuesto.

Política Institucional del IPASME

El área tecnológica también esta contemplada dentro de la política institucional del IPASME, lo cual constituye un aval importante en el desarrollo de la presente investigación:

Modernizar permanentemente los sistemas operativos y administrativos del IPASME mediante una estructura flexible y adecuada a las exigencias cambiantes del medio social, económico y político, que contribuya al fortalecimiento del sector Público en niveles cada vez más altos de eficacia, eficiencia y excelencia en la seguridad y asistencia social de la población afiliada.

Operacionalización de las Variables

Según Balestrini, M. (2002): “Una variable es un aspecto o dimensión de un objeto, o una propiedad de estos aspectos o dimensiones que adquiere distintos valores y por lo tanto varía” (Pág. 113). Por su parte, Tamayo, M. (1999) expone: “Se denomina variable un aspecto o dimensión de un fenómeno que tiene como característica la capacidad de asumir distintos valores, ya sea cuantitativa o cualitativamente”. Este mismo autor enfatiza además que “La validez de una variable depende sistemáticamente del marco teórico que fundamenta el problema y del cual se ha desprendido”.

Por otro lado, Hernández, R. y Otros (2007), refiriéndose a la definición operacional sostiene que: “...especifica qué actividades u operaciones deben realizarse para medir una variable” (Pág. 146).

Con respecto a los factores que deben tomarse en cuenta para la operacionalización de una variable, Tamayo, M. (1999) señala que es necesario tener en cuenta:

La definición nominal, que es simplemente el nombre de la variable que debe medirse.

La definición operacional, que esta compuesta por:

1. Las dimensiones de la variable: son los factores o rangos de la variable que debe medirse.
2. Indicadores de la variable: señalan cómo medir las dimensiones de la variable, por medio de cualidades o propiedades que pueden ser directamente observadas y cuantificadas.
3. Índices: es el resultado de la combinación de los valores obtenidos por un individuo o elemento en cada uno de los indicadores propuestos para medir la variable.

En virtud de que el objetivo general del presente trabajo es: “Plantear una propuesta de alta disponibilidad para el Sistema de Telefonía IP del “Instituto de Previsión y Asistencia Social para los trabajadores del Ministerio del poder popular para la Educación” (IPASME) “, se desprende como variable a operacionalizar: *Alta Disponibilidad para el Sistema de Telefonía IP*, la cual debe medirse en las dimensiones y propiedades (indicadores) señalados en el cuadro 5.

Cuadro 5.

Operacionalización de la variable

Variable	Dimensiones	Indicadores	Items		
			C1	C2	T
Alta Disponibilidad para el Sistema de Telefonía IP	Análisis de los requerimientos en alta Disponibilidad	Fiabilidad		1, 2	
		Recuperación		3, 4	
		Detección de errores		5, 6	
		Análisis de Impacto de Negocios		7, 8, 9, 10	
	Métodos para garantizar la Alta Disponibilidad	Acuerdos de Nivel de Servicio	1, 2		
		Administración de Sistemas	3, 4, 5		
		Redundancia	6		
		Respaldo de recursos críticos	7		
		Estandarización	8, 9, 10		
	Esquemas de Alta Disponibilidad	Características a nivel de hardware	11, 12		13 -15, 17 - 21
		Características a nivel de software			16, 22, 23

Fuente: el Autor

CAPÍTULO III

MARCO METODOLÓGICO

Luego de haber hecho la formulación del problema, así como los objetivos que deben cubrirse, fijar la importancia y el alcance del estudio y asumidas las bases teóricas que marcan el sentido de esta investigación, se seleccionaron los métodos y técnicas utilizados luego en el despliegue de la investigación y que hicieron posible obtener la información requerida.

Dicho de una manera más formal por Balestrini, M. (2002):

El fin esencial del Marco Metodológico, es el de situar en el lenguaje de investigación, los métodos e instrumentos que se emplearán en la investigación planteada, desde su ubicación acerca del tipo de estudio y el diseño de investigación; su universo o población; su muestra; los instrumentos y técnicas de recolección de datos; la medición; hasta la codificación, análisis y presentación de los datos. (Pág. 126).

A continuación se desarrollan los aspectos arriba mencionados para el caso específico de esta investigación.

Tipo de Investigación

Naturaleza de la Investigación

El propósito fundamental de este estudio fue plantear una propuesta de alta disponibilidad para el Sistema de Telefonía IP del “Instituto de Previsión y Asistencia Social para los Trabajadores del Ministerio del Poder Popular para la Educación” (IPASME), caso: Sede Administrativa Principal.

Por consiguiente, la naturaleza de la investigación se corresponde a un Estudio de Proyecto ó Proyecto Factible, apoyándose en la investigación monográfica documental y de campo, lo cual está sustentado en el “Manual para la Elaboración de Trabajo Conducentes a Grado Académico de Especialización, Maestría y Doctorado”, aprobado por el Consejo Universitario de la UCLA en su Sesión N° 1353, (2002), en el cual se indica: “Se entenderá por estudios de proyectos una proposición sustentada en un modelo viable para resolver un problema práctico planteado, tendente a satisfacer necesidades institucionales o sociales y pueden referirse a la formulación de políticas, programas, tecnología, métodos y procesos.” (Pág. 63).

Fases del Estudio

El estudio se desarrolló en base a las tres fases fundamentales de un proyecto factible: Fase I, Estudio Diagnóstico; Fase II, Estudio de Factibilidad y Fase III, Diseño de la Propuesta, según el “Manual para la Elaboración de Trabajo Conducentes a Grado Académico de Especialización, Maestría y Doctorado”, aprobado por el Consejo Universitario de la UCLA en su Sesión N° 1353, (2002).

Estudio Diagnóstico

En esta fase del proyecto se determinó la situación actual de los Sistemas de Telefonía IP con los que cuenta el IPASME, con la finalidad de determinar si existe la necesidad de agregar alta disponibilidad a dichos sistemas. Esta fase de la investigación se desarrolló a través de un estudio de campo donde se recogió la información directamente del personal de la sede administrativa principal de esta institución. A continuación se describen los distintos elementos y acciones desarrollados en la etapa de diagnóstico.

Población y Muestra

En toda investigación es necesario delimitar la población o universo de estudio, para lo cual se debe tener un concepto claro de población o universo en términos de investigación. En este sentido, Balestrini, M. (2002) indica: "...una población o universo puede estar referido a cualquier conjunto de elementos de los cuales pretendemos indagar y conocer sus características, o una de ellas, y para el cual serán válidas las conclusiones obtenidas en la investigación." (Pág. 137).

En el mismo orden de ideas, Hernández, R. y Otros (2007), señala que: "Una población o universo es un conjunto de todos los casos que concuerdan con determinadas especificaciones." (Pág. 239).

En esta investigación, el universo está constituido por la totalidad de los usuarios del Sistema de Telefonía IP de la sede administrativa principal del IPASME, y específicamente, son aquellos empleados registrados como usuarios dentro del Sistema de Telefonía IP, quienes tienen asignada una extensión de dicho sistema. Por lo tanto, esta población o universo de estudio está conformada por un total de 545 personas.

Una vez que se ha situado el universo de estudio, debe determinarse si es posible realizar la recopilación de datos de manera individual sobre todas las unidades que conforman la población o si por el contrario, se requiere introducir técnicas de muestreo para seleccionar un subconjunto representativo del total de la población de estudio.

Según Hernández, R. y Otros (2007): "La muestra es un subgrupo de la población de interés (sobre el cual se recolectarán datos, y que tiene que definirse o delimitarse de antemano con precisión)..." (Pág. 236).

Para la población que fué objeto de estudio en esta investigación, se realizó un muestreo estadístico estratificado, debido a que dentro del universo o población se encontraron dos tipos de usuarios: (a) Usuarios expertos, encargados además de la administración del sistema de Telefonía IP; (b) Usuarios ordinarios, es el resto del

personal distribuido en las distintas oficinas administrativas de la institución. A su vez, para este segundo grupo se utilizó una muestra probabilística, donde todos los elementos tienen la misma posibilidad de ser escogidos.

Hernández, R. y Otros (2007), define una muestra estadística estratificada como “Subgrupo en el que la población se divide en segmentos y se selecciona una muestra de cada segmento”

De la población total de 545 usuarios, 5 corresponden a los usuarios expertos. Debido al bajo número de la población de usuarios expertos, el total de la misma se tomó como muestra, de acuerdo a lo expuesto por Ary, W. (1996): “...si la población posee pequeñas dimensiones, debe ser seleccionada en su totalidad, para así reducir el error en la muestra”

En lo que respecta a los 540 usuarios restantes, se tomó una muestra de tipo probabilística que, tal y como lo señala Hernández, R. y Otros (2007) “Es un subgrupo de la población en el que todos los elementos de ésta tienen la misma posibilidad de ser elegidos”. A continuación se describe detalladamente el procedimiento utilizado para la selección de dicha muestra según este mismo autor.

El tamaño de una muestra de tipo probabilística puede obtenerse a partir de las siguientes ecuaciones:

$$n' = \frac{S^2}{V^2} \quad \text{Ecuación 1} \qquad S^2 = P(1-P) \quad \text{Ecuación 2}$$

$$V^2 = (se)^2 \quad \text{Ecuación 3} \qquad n = \frac{n'}{1 + (n'/N)} \quad \text{Ecuación 4}$$

Donde:

n' : tamaño provisional de la muestra, sin ajustar

S^2 : varianza de la muestra, la cual se determina en términos de la probabilidad

V^2 : varianza al cuadrado de la población

P: probabilidad de ocurrencia del fenómeno. Una muestra es representativa de la población con 95 a 99 % de probabilidad a favor.

se: error estándar, determinado en función de la probabilidad por tanto el error máximo aceptable es de 5 %

N: tamaño de la población

n: tamaño de la muestra

Para el presente objeto de estudio se desea determinar cual es el número de usuarios ordinarios a los que se debe aplicar el instrumento para diagnosticar la necesidad de incorporar Alta Disponibilidad al Sistema de Telefonía IP del IPASME, sabiendo que la población total es de 540, se desea un error estándar no mayor al 3 % y una confiabilidad de 99 %.

De la información anterior se desprenden los siguientes datos:

N = 540 usuarios

se = 0.03

P = 0.9

Sustituyendo los valores correspondientes en la ecuación 2 se determina la varianza de la muestra:

$$S^2 = 0.9(1-0.9) \rightarrow S^2 = 0.09$$

Por otro lado, la varianza al cuadrado de la población se obtiene a partir de la ecuación 3:

$$V^2 = (0.03)^2 \rightarrow V^2 = 0.0009$$

Luego, puede obtenerse el tamaño provisional de la muestra sustituyendo los valores obtenidos en los pasos anteriores dentro de la ecuación 1:

$$n' = \frac{0.09}{0.0009} \rightarrow n' = 100$$

Finalmente, de la ecuación 4 se obtiene el valor de la muestra:

$$n = \frac{100}{1 + (100/540)} \rightarrow n = 84.388$$

Aproximando el valor de n al entero más cercano, se tiene entonces:

$$n = 84$$

Como se observa en el cuadro 6, del total de 540 usuarios ordinarios del Sistema de Telefonía IP, se tomó una muestra de 84, además de los 5 expertos anteriormente mencionados. La distribución total de la población y la muestra se resume en el cuadro 7.

Cuadro 6

Cálculo de la muestra

Parámetro	Simbología	Resultado
Varianza de la muestra	S^2	0,09
Varianza de la población	V^2	0,0009
Tamaño provisional de la muestra	n'	100
Tamaño real de la muestra	n	84

Fuente: el Autor

Cuadro 7

Distribución total de la población y la muestra

Segmentos de la muestra	N	n
Usuarios Expertos	5	5
Usuarios administrativos	540	84
Población y Muestra total	545	89

Fuente: el Autor

Procedimiento Diagnóstico

Para llevar al cabo el estudio diagnóstico de la necesidad de la incorporación de Alta disponibilidad se completaron los siguientes pasos:

1. Elaboración de los instrumentos para la recolección de datos.
2. Validación de los instrumentos elaborados.
3. Determinación de la confiabilidad de los instrumentos.
4. Aplicación de los instrumentos
5. Análisis de los datos recolectados
6. Resultados del diagnóstico en relación a la necesidad de la incorporación de Alta Disponibilidad al actual Sistema de Telefonía IP de la sede administrativa principal del IPASME.
7. Conclusiones del diagnóstico.
8. Recomendaciones del diagnóstico

Técnicas e Instrumentos de Recolección de Datos

Las técnicas de recolección de datos son las distintas formas o maneras de obtener la información. Dicho de una manera formal por Balestrini, M. (2002): “Conjunto de técnicas que permitirán cumplir con los requisitos establecidos en el paradigma científico, vinculados a el carácter específico de las diferentes etapas de este proceso investigativo y especialmente referidos al momento teórico y al momento metodológico de la investigación... ”. Este mismo autor clasifica las técnicas de recolección de datos como: (a) Técnicas relacionadas con el análisis documental de las fuentes bibliográficas (análisis de contenido, observación documental, resumen analítico, análisis crítico); (b) Técnicas operacionales para el manejo de las fuentes documentales(subrayado, fichaje, presentación de índices,

cuadros, gráficos e ilustraciones, presentación de trabajo escrito); (c) Técnicas vivas o de relaciones individuales o de grupos (observación directa, entrevistas orales o escritas).

En función de los objetivos definidos en el presente trabajo de investigación, dada la naturaleza del estudio y en función de los datos que se quieren obtener, se utilizaron como técnicas relacionadas con el análisis documental de las fuentes bibliográficas: la observación documental y resumen analítico, y como técnicas vivas: las encuestas, las entrevistas y la observación directa.

El instrumento diseñado para la recolección de los datos fué el cuestionario, diseñado en base a preguntas cerradas y utilizando una escala tipo Likert de cinco alternativas: *TDA = Totalmente de Acuerdo; DA = De Acuerdo; NEU = Neutral; ED= En Desacuerdo; y TED = Totalmente en Desacuerdo*. En esta escala, una respuesta igual a “TDA” implica que, a juicio del encuestado, el Sistema de Telefonía IP cumple en su totalidad con los requerimientos de alta disponibilidad que se evalúan y una respuesta “TED” indica que la alta disponibilidad no se cumple en lo absoluto.

Más concretamente, se elaboraron los siguientes instrumentos:

1. Cuestionario 01: por medio de la técnica de la entrevista se aplicó el cuestionario 01 al personal técnico de la Coordinación de redes conformado por un total de 05 funcionarios. El anexo B muestra el modelo del cuestionario 01
2. Cuestionario 02: utilizando la técnica de la encuesta se aplicó el cuestionario 02 a la muestra calculada para el universo de 540 usuarios ordinarios del Sistema de Telefonía IP. Dicha muestra estuvo conformada por 84 usuarios los cuales fueron seleccionados de manera aleatoria de las distintas oficinas administrativas de la institución. El modelo del cuestionario 02 se muestra en el anexo C
3. Tabla de recopilación de datos: aplicado por el investigador, por medio de un proceso de observación directa, con el fin de cotejar y validar la información colectada a través de los otros instrumentos aplicados en la investigación. El contenido de esta tabla se muestra en el anexo D.

Validación de los Instrumentos

La validez consiste en la forma en la cual el instrumento seleccionado mide lo que realmente se requiere. Para Balestrini, M. (1998) “la validez es un concepto del cual pueden tenerse diferentes tipos de evidencias relacionadas con el contenido, criterio y con el constructo”. En este sentido, la validez de contenido se refiere al grado en que un instrumento refleja un dominio específico de contenido de lo que se mide y al grado en que una medición se relaciona consistentemente con otras de acuerdo con las hipótesis derivadas teóricamente sobre esa variable, siendo un constructo una variable medida dentro de una teoría o esquema teórico.

Según Díaz, H. (2006) una de las maneras de garantizar la validez de un instrumento es a través del juicio de expertos que comprueben la consistencia de ese instrumento. Partiendo de este señalamiento, para determinar la validez de los instrumentos diseñados en esta investigación se tomó la opinión de tres expertos: un especialista en el área de metodología y dos especialistas en el área de estudio de la investigación:

1. Ing. Msc. Maritza Torres – Profesora Titular adscrita al Departamento de Sistemas del DCYT-UCLA
2. Analista de Sistemas Junior Escalona – Jefe del Departamento de Redes y Telecomunicaciones de la UCLA
3. Ing. Msc. Rafael Valero – Docente de Ing. De Sistemas de UNEFA – Barquisimeto.

Para completar el proceso de validación se le hizo entrega a cada experto la información general de la Investigación acompañada de un formato elaborado exclusivamente para capturar sus opiniones y recomendaciones en cuanto a cada ítems de los tres instrumentos diseñados. El modelo de la información entregada a cada experto se puede apreciar en el anexo E.

Determinación de la confiabilidad de los Instrumentos

Para Díaz, H. (2006): “la confiabilidad viene dada en la medida en que un instrumento arroje siempre los mismos resultados, siendo aplicado en repetidas oportunidades a los mismos sujetos”; acotando Muñiz, J. (1996): el procedimiento más utilizado para evaluar la consistencia interna de un test es el coeficiente Alpha de Cronbach, con rango entre “menos uno” (-1) y “más uno” (+1), determinado por la siguiente fórmula:

$$\alpha = \frac{N}{(N - 1)} * \left(\frac{1 - \sum SI^2}{St^2} \right)$$

En donde:

N = Es el numero de ítems.

$\sum SI^2$ = Sumatoria de la varianza por ítems.

St^2 = Varianza Total.

Para el caso de los cuestionarios 01 y 02, con el propósito de determinar cualquier ambigüedad, antes de aplicar el instrumento definitivo se aplicó una prueba piloto a un pequeño grupo de diez (10) personas.

Según lo indican Hernández, R. y Otros (2007), el índice de confiabilidad cuando se utiliza el Alpha de Cronbach debe ser menor o igual a uno (1) para que el valor indicativo del instrumento posea un alto grado de consistencia interna, lo que indica la exactitud y objetividad en los resultados. Específicamente, los criterios que rigen la confiabilidad del instrumento por medio de este método son los siguientes:

No Confiable: rango de -1 a 0

Confiabilidad baja: rango de 0.01 a 0.49

Confiabilidad Moderada: rango de 0.50 a 0.75

Confiabilidad Fuerte: rango de 0.76 a 0.89

Confiabilidad Alta: rango de 0.90 a 1.0

Una vez aplicados los instrumentos a la prueba piloto, se procedió a utilizar la herramienta estadística Minitab, versión 15.1.30.0, para determinar el valor del Alpha de Cronbach y los resultados obtenidos fueron los siguientes:

Cuestionario 01: $\alpha= 0,8483$

Cuestionario 02: $\alpha= 0,8605$

Con estos resultados, se verifica que ambos cuestionarios tienen una confiabilidad interna con criterio de *Confiabilidad Fuerte*, por lo que se procedió con la aplicación de los cuestionarios a toda la muestra. El detalle del procedimiento arrojado por la herramienta utilizada se muestra en el anexo F.

Técnica para el Análisis de los Datos

Para Herández, A. (2003), “el análisis de resultados debe ser la base sobre la cual se genera el modelo operativo que se aspira proponer para modificar la realidad existente”.

En este trabajo, relativo al Diseño de una Propuesta de Alta Disponibilidad para el Sistema de Telefonía IP del IPASME, se utilizaron algunos métodos estadísticos derivados de la estadística descriptiva a objeto de resumir y comparar las propiedades abstraídas de la variable de estudio, como lo es : La Alta Disponibilidad, a partir de sus dimensiones e indicadores obtenidos por medio de la operacionalización de la variable y los cuales fueron abordados en los diversos ítems contenidos en los instrumentos diseñados para la recolección de los datos.

Una vez obtenidos los datos, se efectuó un estudio estadístico-descriptivo a través de la distribución de frecuencias.

Para tabular los datos, se procedió a diseñar una matriz de indicadores para hacer el análisis de las respectivas alternativas y esto se muestra en tablas estadísticas expresadas en frecuencia y porcentaje.

Análisis de los Resultados

El análisis de los resultados obtenidos en los instrumentos aplicados permitió conocer la necesidad de incorporar Alta Disponibilidad al actual Sistema de Telefonía IP de la sede Administrativa Principal del IPASME. Los resultados se presentan a continuación en frecuencia y porcentaje para su descripción y análisis en función de la operacionalización de la variable.

El cuestionario 01, aplicado a los expertos, agrupa los indicadores relacionados con la dimensión de los Métodos para Garantizar la alta Disponibilidad, mientras que el cuestionario 02 aplicado a los usuarios ordinarios, agrupa los indicadores de la dimensión del Análisis de los Requerimientos de Alta Disponibilidad

Análisis de los Requerimientos de Alta Disponibilidad

Las siguientes tablas de distribución corresponden al análisis de los resultados del cuestionario 02, aplicados a los usuarios ordinarios. Los Items de este cuestionario fueron agrupados de acuerdo a los indicadores con los cuales se relacionan, generando una tabla de distribución de frecuencia por cada indicador en estudio.

Cuadro 8

Distribución de frecuencia sobre la opinión de los usuarios con respecto a la fiabilidad del Sistema de Telefonía IP

Indicador: Fiabilidad

Item 1: El servicio telefónico de la institución siempre está disponible.

Item 2: Todos los servicios de valor agregado proporcionados por el sistema telefónico tales como directorio y correo de voz, siempre están disponibles.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	F	%
1	12	14,29	18	21,43	14	16,66	21	25,00	19	22,62
2	11	13,09	17	20,24	23	27,38	19	22,62	14	16,66

Fuente: el Autor

Como se puede apreciar en el cuadro 8, en cuanto a la fiabilidad que perciben los usuarios del servicio de telefonía IP (Item 1) más del 40% expreso su desacuerdo ó total desacuerdo al respecto, mientras que en referencia a la fiabilidad de los servicios de valor agregado, las opiniones estuvieron marcadas hacia una tendencia neutral y en desacuerdo abarcando mas del 60 % de los encuestados. Esto permite concluir que actualmente el Sistema de Telefonía IP objeto de estudio, no es lo suficientemente fiable para los usuarios ya que no logra alcanzar al menos el 50 % de fiabilidad.

Cuadro 9

Distribución de frecuencia sobre la opinión de los usuarios con respecto a la recuperación ante fallos del Sistema de Telefonía IP

Indicador: Recuperación ante fallos

Item 3: Cuando existen caídas inesperadas del servicio telefónico, siempre son solventadas antes que se requiera utilizar otro medio de comunicación.

Item 4: Cuando existen caídas inesperadas del servicio telefónico, todos los servicios de valor agregado son recuperados en conjunto con el servicio telefónico.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
3	09	10,72	18	21,43	12	14,29	25	29,76	20	23,81
4	10	11,90	14	16,66	22	26,19	24	28,57	14	16,66

Fuente: el Autor

Con respecto a la recuperación ante fallos, claramente se observa en el cuadro 9, que mas del 50% de los usuarios no considera que el actual sistema de Telefonía IP cumpla con este requerimiento.

Cuadro 10

Distribución de frecuencia sobre la opinión de los usuarios con respecto a la detección de errores del Sistema de Telefonía IP

Indicador: Detección de errores

Item 5: El sistema telefónico siempre funciona correctamente.

Item 6: Todos los servicios de valor agregado proporcionados por el sistema telefónico tales como directorio y correo de voz siempre funcionan correctamente.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
5	15	17,86	17	20,24	20	23,81	19	22,62	13	15,48
6	17	20,24	20	23,81	19	22,62	17	20,24	11	13,09

Fuente: el Autor

En el cuadro 10, se refleja una distribución relativamente uniforme entre las distintas opciones, de donde puede inferirse que existen algunos métodos que permiten garantizar el correcto funcionamiento del servicio por un tiempo promedio.

Cuadro 11

Distribución de frecuencia sobre la opinión de los usuarios con respecto al impacto del Sistema de Telefonía IP en las actividades laborales

Indicador: Análisis de Impacto de negocios

Item 7: Cuando existen caídas del servicio telefónico, siempre son solventadas antes que se vean afectadas las actividades laborales

Item 8: Se cuenta con un sistema telefónico alternativo cuando existen caídas del sistema de telefonía IP

Item 9: Las actividades de este departamento funcionan sin inconvenientes cuando existen caídas del sistema telefónico.

Item 10: Cuando el sistema telefónico experimenta caídas inesperadas, los servicios para las extensiones tales como directorio y correo de voz se mantienen funcionando.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
7	10	11,90	19	22,62	16	19,05	22	26,19	17	20,24
8	07	8,33	09	10,72	05	5,95	30	35,71	33	39,28
9	04	4,76	04	4,76	15	17,86	29	34,52	32	38,10
10	03	3,57	05	5,95	17	20,24	27	32,15	32	38,10

Fuente: el Autor

Aun cuando el cuadro 11 muestra que los usuarios están en desacuerdo con la afirmación del Item 7 en mas de 40 % y en los Items 8, 9 y 10 se observa un desacuerdo de mas del 60% con las afirmaciones presentadas, lo que permite inferir,

que el Sistema de Telefonía IP de la institución tiene un importante impacto en las actividades del negocio.

Del análisis de los datos anteriores, se puede observar que el actual Sistema de Telefonía IP de la sede administrativa principal del IPASME, no cumple con los requerimientos de Alta Disponibilidad ya que, en su mayoría, las opiniones de los usuarios no apoyan las afirmaciones expuestas.

Métodos para garantizar Alta Disponibilidad

Las tablas de distribución de frecuencia que se muestran a continuación, corresponden al análisis de los resultados del cuestionario 01, aplicado a los usuarios expertos. De igual manera que en el análisis anterior, los Items de este cuestionario fueron agrupados de acuerdo a los indicadores con los cuales se relacionan, generando una tabla de distribución de frecuencia por cada indicador en estudio.

Cuadro 12

Distribución de frecuencia sobre la opinión de los usuarios expertos en cuanto a la existencia de Acuerdos de Nivel de Servicio para el Sistema de Telefonía IP

Indicador: Acuerdos de Nivel de Servicio.

Item 1: Están formalmente establecidos los Acuerdos de Nivel de Servicio.

Item 2: Los Acuerdos de Nivel de Servicio establecidos para el Sistema de Telefonía IP se cumplen a cabalidad.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
1	--	--	--	--	--	--	1	20,00	04	80,00
2	--	--	--	--	--	--	1	20,00	04	80,00

Fuente: el Autor

De la información recopilada en los Items 1 y 2 en el cuadro 12, se puede deducir claramente que no existen Acuerdos de Nivel de Servicio establecidos para el Sistema de Telefonía IP de la institución.

Cuadro 13

Distribución de frecuencia sobre la opinión de los usuarios expertos con respecto a la administración del Sistema de Telefonía IP

Indicador: Administración de sistemas

Item 3: Existe un a planificación estratégica en cuanto al mantenimiento del Sistema de Telefonía IP de manera que el servicio no se vea interrumpido

Item 4: En caso que el Sistema de Telefonía IP experimentara alguna parada no planificada, se cuenta con un plan de contingencia para la recuperación del servicio.

Item 5: Están claramente identificados aquellos elementos que comprometen el funcionamiento del sistema de Telefonía en caso de experimentar alguna falla.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
3	--	--	2	40,00	--	--	2	40,00	1	20,00
4	--	--	--	--	--	--	2	40,00	3	60,00
5	3	60,00	2	40,00	--	--	--	--	--	--

Fuente: el Autor

Como se puede apreciar en el cuadro 13, en cuanto a la administración del Sistema de Telefonía IP para garantizar su disponibilidad, se observa que los usuarios expertos tienen identificados los puntos simples de fallos (Item 5), sin embargo no se cuenta con planes de contingencia para la recuperación del sistema (Item 4) y en cuanto a la planificación estratégica de los mantenimientos del sistema (Item 3) existe contradicción entre los entrevistados por lo que se puede inferir que esta planificación no está siendo divulgada oportunamente. Esto permite concluir que

actualmente existen algunos parámetros de administración del sistema de Telefonía IP, pero no se integran adecuadamente para garantizar la Alta Disponibilidad del mencionado sistema.

Cuadro 14

Distribución de frecuencia sobre la opinión de los usuarios expertos con respecto a la redundancia y respaldo de recursos críticos del Sistema de Telefonía IP

Indicadores: redundancia y respaldo de recursos críticos

Item 6: El Sistema de Telefonía IP cuenta con una infraestructura redundante para proteger los puntos de fallo.

Item 7: Se cuenta con políticas de respaldo permanentes para el respaldo de las bases de datos utilizadas por el Sistema de Telefonía IP.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
6	--	--	--	--	--	--	--	--	5	100,00
7	--	--	--	--	--	--	5	100,00	--	--

Fuente: el Autor

Se evidencia en la información presentada en el cuadro 14, que no existe redundancia (Item 6) ni tampoco se cuenta con políticas de respaldo de recursos críticos del sistema (Item 7).

Cuadro 15

Distribución de frecuencia sobre la opinión de los usuarios expertos con respecto a la estandarización del Sistema de Telefonía IP

Indicador: Estandarización

Item 8: La infraestructura telefónica es homogénea en cuanto al tipo de centrales IP que prestan el servicio.

Item 9: La infraestructura telefónica es interoperable en cuanto a las distintas centrales IP que prestan el servicio.

Item 10: Todos los terminales ó teléfonos IP de los usuarios utilizan el mismo protocolo de comunicación con las centrales IP instaladas.

ITEM	Totalmente de Acuerdo		De acuerdo		Neutral		En Desacuerdo		Totalmente en Desacuerdo	
	f	%	f	%	f	%	f	%	f	%
8	--	--	--	--	--	--	2	40,00	3	60,00
9	--	--	5	100	--	--	--	--	--	--
10	--	--	--	--	--	--	3	60,00	2	40,00

Fuente: el Autor

Como se puede apreciar en el cuadro 15, en cuanto a la estandarización tanto a nivel de cliente (Item 10) como servidor (Item 8), los usuarios expertos coinciden en la negación de la afirmación presentada lo que indica una alta de estandarización dentro del sistema. Sin embargo, si existe interoperabilidad (Item 9) Esto permite concluir que actualmente el Sistema de Telefonía IP no cuenta con una infraestructura completamente estandarizada.

El cuestionario 01 aplicado a los usuarios expertos, también incluye 2 últimos Items (Item 11 e Item 12) los cuales corresponden a la dimensión referente a los esquemas de Alta Disponibilidad, y abordan el indicador correspondiente a las características a nivel de hardware. Sin embargo, estos Items se utilizaron como complemento al tercer instrumento aplicado como lo fue la tabla de recolección de datos. Esta última información más que ser utilizada para el diagnóstico, es relevante para el diseño de la propuesta.

Del análisis realizado en función de cada uno de los indicadores de las dimensiones de la variable de estudio, se puede concluir que el actual Sistema de Telefonía IP de la sede administrativa principal del IPASME, no incorpora los

métodos para garantizar Alta disponibilidad tales como Acuerdos de Nivel de servicio, Redundancia, respaldo de recursos críticos ni estandarización.

Conclusiones del diagnóstico

1. En cuanto a los requerimientos en Alta Disponibilidad, el actual Sistema de Telefonía IP presenta deficiencias en cuanto a fiabilidad, recuperación y detección de errores.
2. La disponibilidad del Sistema de Telefonía IP compromete considerablemente la continuidad de las actividades en la institución de acuerdo al análisis del impacto en el negocio.
3. No están establecidos los Acuerdos de Nivel de Servicio para el Sistema de Telefonía IP de la institución.
4. No se cubren todos los parámetros de administración del sistema que permitan garantizar la disponibilidad del servicio telefónico.
5. Actualmente el sistema de Telefonía IP de la institución no incorpora métodos como la redundancia, respaldo de recursos críticos y estandarización para garantizar la Alta Disponibilidad.

Las conclusiones antes expuestas concuerdan con la observación directa que llevó a cabo el investigador durante el proceso de recolección de datos, permitiendo concluir que: ***“El Sistema de Telefonía IP de la sede administrativa principal del IPASME No incorpora elementos de Alta disponibilidad que permitan garantizar la continuidad de los servicios prestados por dicho sistema”.***

Recomendaciones del diagnóstico

En base al análisis de los datos obtenidos se determinó que el Sistema de Telefonía IP de la sede administrativa Principal del IPASME debería ser mejorado de

tal manera que permita garantizar la Alta Disponibilidad de los servicios de Telefonía IP.

Por lo tanto, se recomienda incorporar un esquema de Alta Disponibilidad al actual Sistema de Telefonía IP de la institución que permita garantizar su continuidad operativa. El esquema propuesto debe integrarse con el actual Sistema de Telefonía IP incorporando los métodos en alta disponibilidad estudiados en el marco teórico de esta investigación y adaptándolos al caso particular de Sistemas de Telefonía IP.

En resumen, el esquema de Alta Disponibilidad a incorporar debe tomar en cuenta que:

- El sistema esta compuesto por dos modelos de centrales telefónicas: una propietaria y una basada en software libre. No es tema de esta investigación incorporar Alta disponibilidad a la central propietaria.
- Debe proponerse un esquema redundante orientado a software por encima de esquemas redundantes orientados a hardware.
- Por tratarse de una institución pública, se debe emplear prioritariamente software de libre distribución para las soluciones de software que se propongan
- Debe considerarse la escalabilidad del actual Sistema de Telefonía IP de la institución.

Estudio de Factibilidad

En esta segunda fase según Hernández, R. (2007) "... es donde se establecen los criterios que permiten asegurar el uso óptimo de los recursos empleados así como los efectos del proyecto en el área o sector al que se destina". En este sentido, los estudios que permitieron determinar de manera objetiva la viabilidad del diseño propuesto corresponden al estudio técnico, que analiza lo referente a la capacidad para desarrollar la propuesta, el estudio operativo, que permite predecir si se pondra en marcha el sistema propuesto, el estudio de los recursos económicos que permitan garantizar el desarrollo de la propuesta, y el estudio institucional que sirva como aval

para soportar la futura ejecución de la propuesta que se genere. A continuación se describe cada uno de estos elementos.

Factibilidad Técnica

La factibilidad técnica consistió en recolectar la información sobre los componentes técnicos que serían necesarios al momento de implantar el diseño propuesto. Estos componentes no sólo corresponden a los elementos de hardware y de software necesarios, sino también a la capacidad del recurso humano con que cuenta la institución para implantar, administrar y mantener el sistema propuesto. En este sentido, se completó un estudio técnico tomando en cuenta tres tipos de elementos:

1. Hardware: El hardware necesario para el momento de la implantación del diseño propuesto está conformado por los dos servidores que se detallan en la descripción de la propuesta (ver cuadro 18 pag. 92) y la conexión de estos equipos a la red Interna de la sede administrativa principal del IPASME mediante una red de topología de estrella, utilizando cable par trenzado sin apantallamiento (UTP) categoría 6 ó superior, según las normas internacionales del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). Evaluando los equipos actualmente dispuestos para el funcionamiento de la central telefónica basada en software libre, se requiere que la institución invierta en la adquisición de al menos dos (02) servidores con las especificaciones técnicas señaladas. Según información proporcionada por la Oficina Central de Tecnología y Sistemas de Información del IPASME, actualmente se encuentran en proceso de licitación para la adquisición de nuevos equipos de computación, entre los cuales se encuentran equipos que cubren los requerimientos mínimos antes expuestos y podrán ser dispuestos para la implantación del diseño propuesto en esta investigación.

Con respecto la interconexión física de los servidores a la red interna de la sede administrativa principal del IPASME, la institución cuenta con una red de

área local (LAN) que cubre los requerimientos para la instalación de los servidores requeridos.

2. Software: El hardware necesario para la implantación del diseño propuesto debe tener instalado como Sistema Operativo base Linux Debian 6.0 ó superior, sobre el cual debe instalarse el software de central de telefonía IP Asterisk y los componentes del proyecto Linux-HA, todos de libre distribución por lo que pueden ser descargados de los repositorios dispuestos para tales fines en cualquier momento y sin ninguna limitación, por lo tanto no constituyen un obstáculo en la implementación del diseño propuesto.
3. Experiencia técnica: La coordinación de redes y comunicaciones adscrita a la Oficina Central de Tecnología y Sistemas del Información del IPASME, tiene dispuesto personal técnico especializado en el actual Sistema de Telefonía IP instalado en la institución. Este personal cuenta con el conocimiento necesario en el Sistema Operativo base y el software de la central telefónica, y por medio de la documentación proporcionada en esta investigación puede comenzar su entrenamiento dentro del proyecto LINUX-HA, el cual puede ser complementado con una gran cantidad de información proporcionada por la comunidad que hace vida dentro de este proyecto.

Factibilidad Operativa

Este estudio de factibilidad permitió determinar si el diseño propuesto funcionará al momento que la institución decida implantarlo. Este estudio involucra la disponibilidad y/o la resistencia que pudieran hacer tanto los usuarios como los administradores del sistema de Telefonía IP al nuevo diseño propuesto, ya sea por su complejidad o por desconocimiento en cuanto a su forma de trabajar.

El diseño propuesto no involucra cambios que puedan ser percibidos por los usuarios en cuanto al uso cotidiano del Sistema de Telefonía IP. Por otro lado, de lo percibido en el proceso de levantamiento de información, los administradores

del sistema de Telefonía IP tienen la mayor disposición de adiestrarse en los nuevos elementos introducidos en el diseño propuesto, ya que facilitaría sus actividades de administración, control y mantenimiento de dicho sistema.

Factibilidad Institucional

La Oficina Central de Tecnología y Sistemas de Información del IPASME tiene entre sus políticas: “Modernizar permanentemente los sistemas operativos y administrativos del IPASME mediante una estructura flexible y adecuada a las exigencias cambiantes del medio social, económico y político, que contribuya al fortalecimiento del sector Público en niveles cada vez más altos de eficacia, eficiencia y excelencia en la seguridad y asistencia social de la población afiliada.”

En el marco de esta política se incluyen sus sistemas de comunicaciones, particularmente el Sistema de Telefonía IP de la sede administrativa principal de esta institución. Además, por tratarse de una institución pública se debe adoptar prioritariamente el uso de Tecnologías libres en sus sistemas de información y comunicación, y es justo en este punto donde el diseño propuesto actúa para contribuir a la incorporación de las nuevas tendencias tecnológicas en la institución lo que garantiza la factibilidad institucional del diseño propuesto.

Factibilidad Económica

La inversión económica que deberá hacer la institución al momento de implantar el diseño propuesto se reduce a la adquisición de los dos servidores descritos en el diseño de la propuesta. El costo aproximado para un servidor con las características señaladas es de: Bsf. 27.000,00. Ya que se recomienda adquirir dos equipos iguales, el costo total de la inversión está en el orden de Bsf. 54.000,00

En cuanto a la inversión en materia de software, esta es nula ya que todas las aplicaciones manejadas en el diseño propuesto son desarrolladas bajo estándares abiertos y de libre distribución.

Finalmente, la inversión en capital humano se limita al personal que administra actualmente el Sistema de Telefonía IP perteneciente a la institución. Para la ejecución, mantenimiento y administración del diseño propuesto no se hace necesario incorporar nuevo personal.

Por todo lo antes expuesto, la factibilidad económica del diseño propuesto estaría garantizada sólo con la adquisición de los servidores, ya que el resto de los elementos que lo componen, no representan gastos adicionales para la institución.

CAPITULO IV

PROPUESTA DEL ESTUDIO

En esta sección del trabajo, se procedió a evaluar la incorporación de los diversos métodos de Alta disponibilidad estudiados en el marco teórico de esta investigación al Sistema de Telefonía IP en estudio. Luego se procedió a seleccionar el esquema que mejor se adapta a los requerimientos detectados en la fase de diagnóstico para posteriormente presentar el diseño propuesto y finalmente se construyó un escenario que permitió la evaluación del diseño propuesto.

Esquemas de Alta Disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME

Como punto de partida para el diseño de los esquemas propuestos, se consideró el hecho de proporcionar Alta Disponibilidad tanto a nivel de servicios como a nivel de datos. Además, estos esquemas deben cubrir los requerimientos detectados en la fase de diagnóstico, así como las recomendaciones realizadas al final de esa misma fase. Partiendo de estas premisas, fueron evaluados los escenarios de Alta Disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME que se muestran a continuación.

En la figura 16 se muestra el esquema actual del Sistema de Telefonía IP de la sede administrativa principal del IPASME, compuesto por una central IP propietaria marca Siemens, modelo HiPath 4000 y un servidor Asterisk el cual se ejecuta sobre Debian 6.0 como sistema operativo base. El equipo Siemens además de atender los servicios de Telefonía IP a los usuarios de las extensiones directamente conectadas a ella, también sirve de pasarela para la comunicación externa de los usuarios de las extensiones configuradas en la central IP Asterisk. Las dos centrales se comunican por medio de un enlace troncal, el cual fue configurado por el personal encargado de la administración del Sistema de Telefonía IP de la institución.

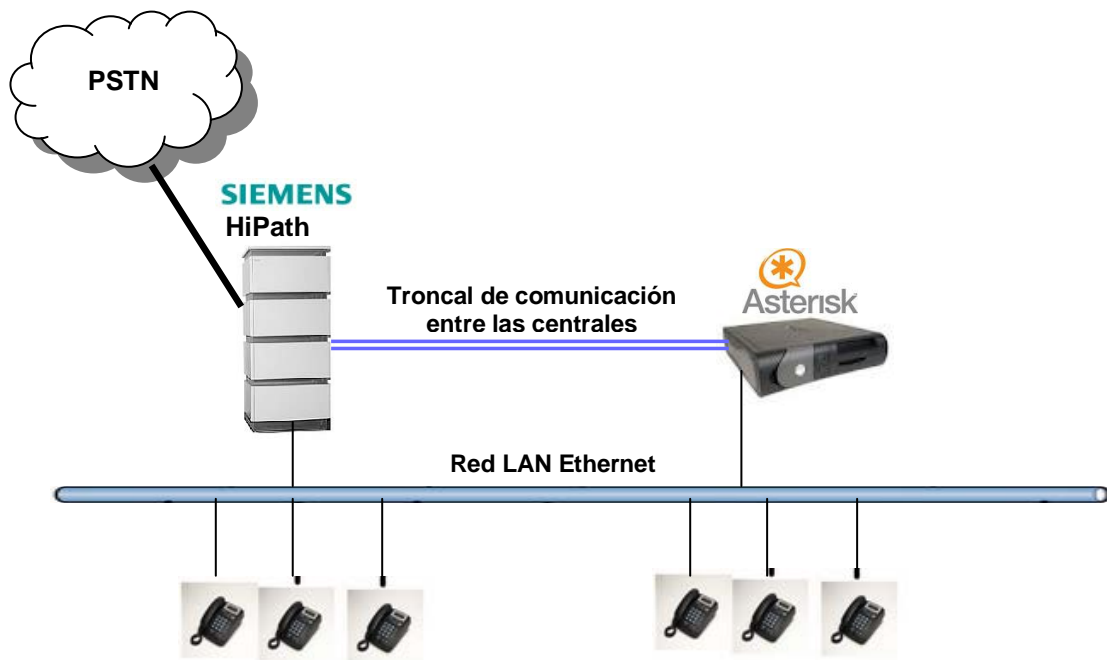


Figura 16. Arquitectura actual del Sistema de Telefonía IP de la sede administrativa principal de IPASME
Fuente: OCTSI

De acuerdo a las recomendaciones de la fase de diagnóstico se tiene que los esquemas de alta disponibilidad propuestos van dirigidos al segmento del sistema conformado por el servidor Asterisk, el cual está instalado actualmente sobre un

computador de escritorio marca Dell, modelo GX 280. En este segmento se pueden identificar los siguientes puntos de fallo: a). El servicio telefónico ofrecido por la PBX Asterisk; b). La información de los sistema de ficheros de Asterisk que permiten autenticación, autorización y almacenamiento de toda la información de usuarios.

De lo anterior surge el primer escenario propuesto, el cual incorpora redundancia al principal punto de fallo que puede identificarse: la central telefónica. Como lo muestra la figura 17, se trata de un esquema basado en un cluster de alta disponibilidad conformado por dos nodos configurados idénticamente (tanto a nivel del servicio de Asterisk, como de la troncal que comunica a las dos centrales).

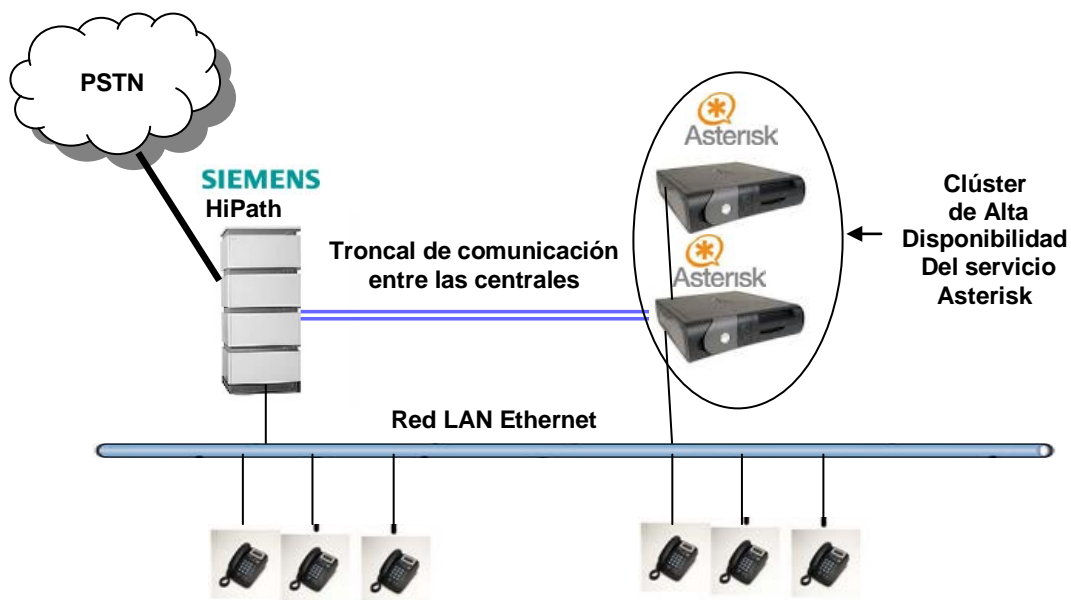


Figura 17. Esquema 1. Propuesta básica de Alta disponibilidad en un sistema de Telefonía IP

Fuente: El autor

Para que la configuración de Asterisk pueda mantenerse igual en todos los nodos y en todo momento, se hace necesario el uso de algún software de alta disponibilidad a nivel de servicios. En base a la revisión teórica que se hizo en esta

investigación, se comparan en el cuadro 16 los diversos proyectos orientados a ofrecer alta disponibilidad a nivel de los servicios.

Cuadro 16

Ventajas y desventajas de los distintos proyectos de Alta disponibilidad de servicios

Software	Ventajas	Desventajas
HA-OSCAR	Su objetivo es proporcionar de manera combinada el poder de la alta disponibilidad con un gran rendimiento de cómputo	Actualmente es utilizado en pocas industrias y su comunidad no se encuentra activa por lo que no hay buen soporte.
The High Availability Linux Project (Proyecto Linux –HA)	Es una de las soluciones más ampliamente extendidas. Se integra perfectamente con multitud de servicios y sigue mejorando gracias a la aportación de su activa comunidad	Poca documentación en español
Kimberlite	Es una completa infraestructura de alta disponibilidad de código abierto que incluye garantía sobre la integridad de los datos.	Actualmente es un proyecto abandonado o de muy baja actividad
LifeKeeper	Es una alternativa moderna y flexible en	Es un producto propiedad de la empresa <i>SteelEye</i>

	cuanto a configuración, comparada con los productos que ofrecen soluciones de alta disponibilidad.	
HP Serviceguard	Es un software especializado para ofrecer alta disponibilidad a las aplicaciones más críticas ante fallos tanto hardware como software	Inicialmente estuvo disponible tanto para Linux como para HP-UX. Sin embargo, hace algún tiempo HP anunció que el producto sería discontinuado para Linux
Red Hat Cluster Suite	Proporciona, además de alta disponibilidad, balanceo de carga. Dispone de soluciones listas para implantar para la mayoría de los servicios, pudiendo además proporcionar soporte cuando sea necesario.	Ha sido diseñado específicamente para Red Hat Enterprise Linux

Fuente: El autor

Una vez analizadas las características de los distintos proyectos expuestos en el cuadro 16, se llegó a la conclusión que la solución de software que mejor se adapta al sistema de telefonía IP de este estudio, según los requerimientos y recomendaciones acotados en la fase de diagnóstico, es el proyecto de LINUX-HA, ya que es una solución de software libre que se integra perfectamente con multitud

de servicios, es confiable al ser una de las soluciones más utilizadas actualmente y es perfectible al estar respaldada por una comunidad activa, lo cual lleva al proyecto a una constante evolución.

Con la integración del primer esquema propuesto y el software de Alta disponibilidad de servicios del proyecto LINUX-HA, se solventa el punto de fallo más importante como lo es el servicio de Asterisk Sin embargo, el punto de fallo relacionado con la integridad de la información de configuración y usuarios de Asterisk sigue existiendo.

Una solución que permite incorporar Alta disponibilidad de datos, para eliminar el punto de fallo de los ficheros de información de configuración al esquema anterior, se muestra en la figura 18.

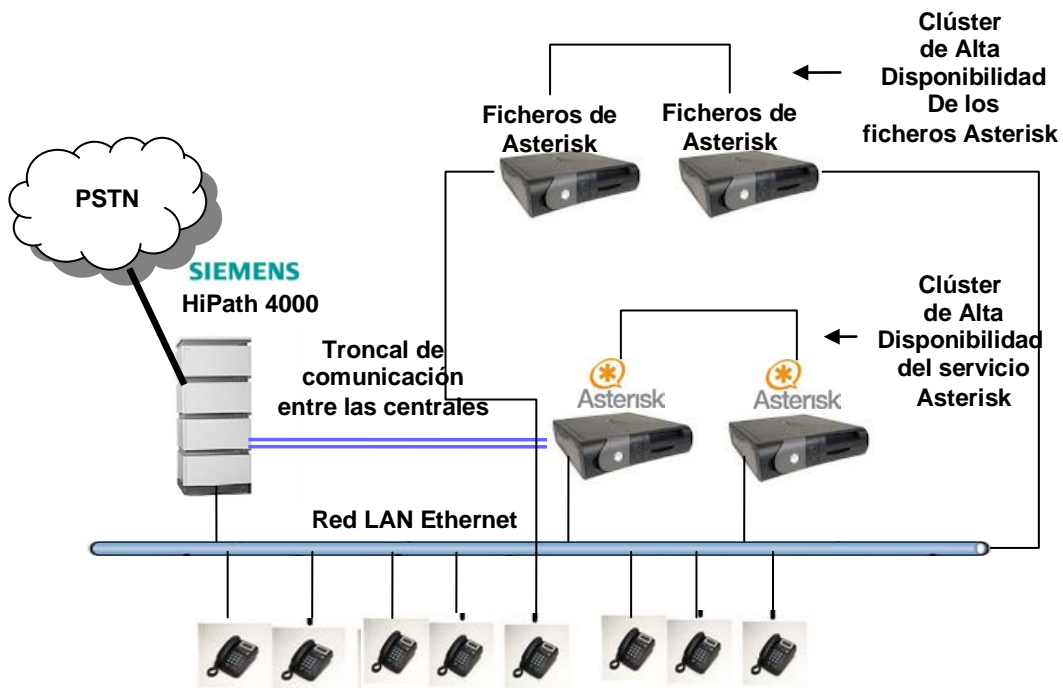


Figura 18. Esquema 2. Propuesta de Alta disponibilidad de servicios integrada con alta disponibilidad de datos

Fuente: El autor

En este segundo esquema, se combina un cluster de Alta Disponibilidad en el servicio Asterisk con un cluster de Alta Disponibilidad en datos, en el cual se separa físicamente la base de datos de usuarios del servicio de Asterisk. De esta manera, si ocurriera un fallo en el nodo que está prestando actualmente el servicio de Asterisk, la información de configuración y usuarios seguiría disponible para el resto de los nodos que forman parte del cluster de alta disponibilidad de servicio.

En este caso, también es necesario incorporar elementos de software que permitan mantener la replicación de los bloques de datos en tiempo real, considerándose para ello los proyectos presentados en la revisión teórica de esta investigación. En el cuadro 17 se exponen las ventajas y desventajas de cada uno de ellos.

Cuadro 17

Ventajas y desventajas de los distintos proyectos de Alta disponibilidad de datos

Software	Ventajas	Desventajas
DRBD	Replicación en tiempo real de manera continua, transparencia en las aplicaciones que estén almacenando datos en la unidad, posibilidad de recuperación de los datos ante un desastre. Se complementa con el proyecto de alta disponibilidad en servicios <i>LINUX-HA</i>	La mayor parte de la documentación está en inglés.
FreeNAS	Está basado en <i>FreeBSD</i>	Actualmente existe muy

	que es un derivado de <i>Unix</i> , permitiendo clientes tanto de <i>GNU/Linux</i> como <i>Windows</i> y por supuesto <i>Unix</i> .	poca documentación y comunidades activas.
Openfiler	Provee buena integración con <i>iSCSI</i> (Internet <i>SCSI</i>) y “ <i>fibre channel</i> ”, muy útiles en entornos empresariales y en virtualización tal como <i>Xen</i> y <i>VMware</i> .	La versión que incorpora Alta Disponibilidad no es de libre distribución.
MySQL Replication	Consiste en una replicación asíncrona unidireccional (un servidor actúa como maestro y uno o más actúan como esclavos).	No se realizan las copias de las bases de datos para mantenerlas sincronizadas. En su lugar se informa de las operaciones realizadas en el servidor maestro. No es posible realizar cambios en los servidores esclavos, son exclusivamente para consultas.

Fuente: El autor

De los proyectos de alta disponibilidad en datos descritos en el cuadro 17, el que ha sido probado exitosamente la mayor cantidad de veces con la propuesta de alta disponibilidad en servicios de LINUX-HA, es el proyecto DRBD. Por lo tanto, se

propone su utilización como software de alta disponibilidad para la replicación de los bloques de datos de usuarios dentro de la propuesta diseñada en este trabajo.

El esquema 2 planteado en la figura 18, representa un esquema ideal para la incorporación de Alta disponibilidad para el Sistema de Telefonía IP de la sede administrativa principal del IPASME. Sin embargo, dicho esquema puede ser optimizado en función de un mayor aprovechamiento de los recursos de hardware ya que, en lugar de utilizar dos nodos para configurar alta disponibilidad en el servicio Asterisk y otros dos nodos para configurar la alta disponibilidad en datos, se pudieran unir ambas soluciones en sólo dos nodos. En la figura 19 se muestra el esquema 3, donde se configuran dos nodos que ofrecen alta disponibilidad en servicios y alta disponibilidad en datos, en forma integrada.

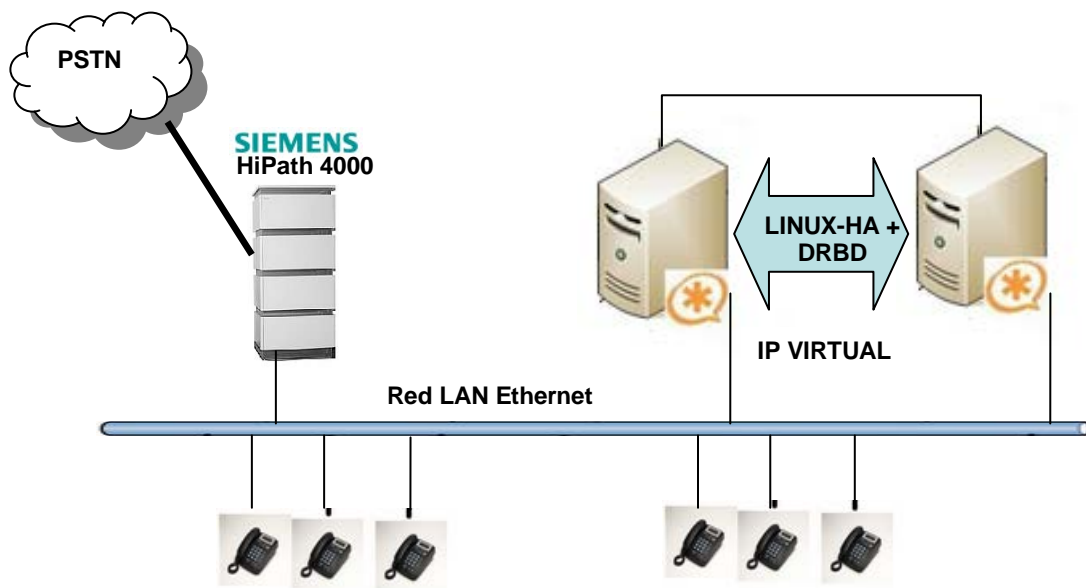


Figura 19. Esquema 3. Propuesta integrada de Alta disponibilidad en servicios y datos para el Sistema de Telefonía IP de la sede administrativa principal del IPASME

Fuente: El autor

Descripción de la propuesta

Partiendo del esquema general propuesto en la figura 19, a continuación se detallan los elementos tanto a nivel de hardware como de software que forman parte del diseño de Alta Disponibilidad propuesto para el Sistema de Telefonía IP de la sede administrativa principal del IPASME.

Arquitectura del Hardware

En cuanto a las especificaciones técnicas del hardware a utilizar al momento de implantar la propuesta diseñada, utilizando como base la guía de requisitos del sistema elaborada por Guadamuz, E. (ver cuadro 3, Pág. 25.), se recomienda hacer uso de dos (02) equipos, los cuales deben cumplir con las especificaciones mínimas que se muestran en el cuadro 18.

Cuadro 18

Arquitectura de hardware para cada uno de los nodos del clúster

Descripción del producto	Especificaciones técnicas mínimas requeridas
Número de procesadores	1
Núcleo del procesador	Dual
Velocidad del procesador	3.2 GHZ
Capacidad de memoria RAM instalada	4 GB
Tipo de memoria	DDR2
Cantidad de discos duros	2
Capacidad por disco	160 GB
Velocidad del disco	10000 rpm
Cantidad de adaptadores de red	2
Tecnología de los adaptadores de red	Gigabit Ethernet

Tipo de conector de red	RJ-45
Cantidad de fuentes de poder	2
Tipos de fuentes de poder	Redundantes
Potencia de fuentes de poder	800 W
Voltaje de fuentes de poder	100 a 240 V a 60 Hz

Fuente: El autor

Arquitectura del Software

Luego de establecer la arquitectura de hardware mínima para la propuesta, el siguiente paso es detallar los componentes de la arquitectura del software que debe instalarse en los equipos especificados.

El Sistema Operativo

En virtud de que el objetivo general de esta investigación es incorporar Alta Disponibilidad al actual Sistema de Telefonía IP de la sede administrativa principal del IPASME, suena lógico pensar en mantener la misma distribución de Linux que se encuentra actualmente instalada : Debian 6.0, el cual es considerado una de las distribuciones de Linux más estable para la instalación de servicios empresariales y de red y soporta muy bien DRBD y LINUX-HA. Por otra parte, hay miles de páginas y de foros donde se puede conseguir ayuda en el caso de tener problemas de instalación, configuración o uso de esta distribución.

Soporte RAID1

Se propone configurar los discos duros con el soporte RAID1 por software de DEBIAN, lo cual va a permitir que los discos duros de cada nodo contengan cada uno de ellos una copia íntegra de todo el sistema. Un RAID1 crea una copia exacta (o

espejo) de un conjunto de datos en dos o más discos, lo que resulta útil cuando el rendimiento en lectura es más importante que la capacidad.

Un RAID1 clásico consiste en dos discos en espejo, lo que incrementa exponencialmente la fiabilidad respecto a un sólo disco; es decir, la probabilidad de fallo del conjunto es igual al producto de las probabilidades de fallo de cada uno de los discos (pues para que el conjunto falle es necesario que lo hagan todos sus discos). Adicionalmente, dado que todos los datos están en dos o más discos, con hardware habitualmente independiente, el rendimiento de lectura se incrementa aproximadamente como múltiplo lineal del número de copias; es decir, un RAID1 puede estar leyendo simultáneamente dos datos diferentes en dos discos diferentes, por lo que su rendimiento se duplica y el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda y subiendo la tasa de transferencia, con el único límite de la velocidad soportada por la controladora RAID.

Soporte para Alta disponibilidad en datos

Tal y como se determinó en la primera parte de este capítulo, la arquitectura de alta disponibilidad en datos para la propuesta consiste en la configuración de un cluster de alta disponibilidad basado en la solución de software DRBD (por sus siglas en inglés *Distributed Replicated Block Device*). DRBD es un *software* de replicación de dispositivos de bloque (discos duros, particiones, volúmenes, etc.) que permite formar un *RAIDI* a través de la red.

Según Méndez, F. (2009), quien extrae las principales características de *DRBD del Manual de usuario de la página oficial de esta herramienta* son:

- **Tiempo real.** La replicación se realiza de manera continua.
- **Transparencia.** Las aplicaciones que estén almacenando datos en la unidad no sabrán que esta se está replicando en varias localizaciones.

- **Síncrono o asíncrono.** Con replicación síncrona la aplicación que escribe es notificada que la escritura se ha completado solo después de que ésta haya sido replicada. Con replicación asíncrona la notificación de que la escritura se ha completado será entregada a la aplicación cuando ésta haya sido realizada de manera local, esto es, antes de que la escritura haya sido propagada al resto de ordenadores.

DRBD ha sido implementado como un módulo adicional para el *Kernel Linux* que permite crear dispositivos de bloques virtuales por lo que está muy próximo al sistema de *E/S*. Es por ello por lo que posee una extrema flexibilidad y versatilidad convirtiendo a *DRBD* en una buena solución para la redundancia de datos mediante la replicación estos.

DRBD requiere definir los recursos de datos a los cuales se va a dotar de alta redundancia. Más detalladamente, un recurso en *DRBD* es una colección de términos que hace referencia a todos los aspectos de un dispositivo particular de almacenamiento replicado. Un recurso está formado por:

- Nombre. Un nombre para el recurso.
- Unidad *DRBD*. Disponer de un dispositivo o unidad de bloque virtual. Se encuentran en */dev/drbdm* donde $0 < m < 147$.
- Configuración del disco. Relacionado con todos los aspectos de copia local de los datos y *meta datos* de uso interno de *DRBD*.
- Configuración de red. Relacionado con todos los aspectos de comunicación con el nodo con el que se van a replicar los datos.

En *DRBD* cualquier recurso tiene un rol que puede ser primario o secundario. Estos roles hacen referencia al concepto de disponibilidad del almacenamiento. Sin embargo, hay otros dos términos que generan mucha confusión con estos: activo y pasivo. Estos últimos hacen referencia a la disponibilidad de una aplicación.

Usualmente en entornos de alta disponibilidad el nodo primario es también el nodo activo, pero esto no tiene por qué ser siempre así.

- Una unidad o recurso *DRBD* con papel de primario se puede usar sin restricciones para operaciones de lectura y escritura.

· Una unidad o recurso *DRBD* con el papel de secundario, solo se encarga de recibir todas las actualizaciones procedentes del recurso primario no permitiéndosele el acceso a los datos para lectura o escritura. De esta forma se mantiene la coherencia de caché.

Los roles de un recurso se pueden modificar ya sea manualmente o de manera automática por una aplicación de gestión del cluster *DRBD*. Modificar el rol de un recurso en un nodo de secundario a primario se conoce como “ascenso o promoción”, mientras que modificar el rol de primario a secundario se conoce como “descenso”.

Modos de funcionamiento de DRBD (Méndez, F., 2009):

DRBD por defecto se encuentra configurado en modo *single-primary*, de tal manera que cualquier recurso solamente puede tomar el rol o papel de primario en uno de los nodos. Ello permite garantizar que sólo uno de los nodos del cluster podrá manipular los datos (el primario) mientras que la funcionalidad del otro nodo se limita a escribir los datos que recibe para mantenerlos así replicados. Este modo es, por tanto, compatible con sistemas de ficheros como *ext3*, *ext4*, *XFS*, etc. Sin embargo, a partir de la versión de *DRBD 8.0*, se ha añadido un nuevo modo de funcionamiento conocido como modo *dual-primary* en el que cualquier recurso puede tener en ambos nodos del cluster el rol de primario. Esto es posible desde que se dispone sistemas de almacenamiento compartido concurrente tales como *GFS* y *OCFS2* que permiten la modificación de ficheros de forma concurrente.

Cada uno de estos modos de redundancia de datos puede tener un enfoque para ser integrado con alta disponibilidad o con alto rendimiento. El modo *single-primary* posee un enfoque de alta disponibilidad mientras que el modo *dual-primary* posee un enfoque de alto rendimiento, ya que permite el acceso concurrente a los datos por parte de los dos nodos.

Instalación de DRBD:

Tomando como base la guía de usuario disponible en la página oficial del proyecto DRBD: <http://www.drbd.org/user-guide/>, el primer paso es particionar ambos servidores exactamente igual y crear una partición extra en la que se van a tener los datos replicados. Esta partición extra debe ser ext3 o ext4 y debe dejarse sin montar.

Los diversos paquetes para distribuciones como *Debian*, *Ubuntu*, *SUSE Enterprise* y *RHEL* entre otras, se pueden encontrar en la página oficial del proyecto. Drbd es muy fácil instalar actualmente ya que viene compilado de serie en las versiones recientes. Sólo hay que cargarlo con modprobe o modconf:

```
# modprobe drbd
```

Luego deben instalarse las utilidades que lo componen (drbdadm, drbdsetup, drbdmeta) usando el siguiente comando:

```
# aptitude install drbd8-utils
```

Los pasos que se acaban de realizar, se han de llevar a cabo en ambos nodos.

El archivo de configuración se encuentra en /etc/drbd.conf .Este debe de ser idéntico en los dos nodos:

```
resource nombre_recurso { # Por ejemplo: resource r0
common {
protocol C;                # Usado para replicación sincrónica
}
on nombre_del_nodo_1 {
device /dev/drbd0;         # Ruta de la unidad virtual /dev/drbdx
disk /dev/sdb1;           # Ruta de la partición a usar para replicación
address direcciónIP_nodo_1; #dirección IP asignada al nodo
puerto_para_replicar;
meta-disk internal;
```

```

}
on nombre_del_nodo_2 {
device /dev/drbd0;          # Ruta de la unidad virtual /dev/drbdx
disk /dev/sdb1;           # Ruta de la partición a usar para replicación
address direcciónIP_nodo_2; #dirección IP asignada al nodo
puerto_para_replicar;
meta-disk internal;
}
}

```

Ahora es necesario, *en ambos nodos*, crear el espacio para metadatos y la unidad virtual a la que se ha hecho referencia en el archivo de configuración *drbd.conf*:

```

#drbdadm create-md nombre_recurso
#drbdadm attach nombre_recurso

```

NOTA: Si la salida de *create-md* no es la esperada como si se obtiene un error al ejecutar *attach*, el motivo es que el servicio *drbd* esta arrancado y no permite ni crear los *metadatos* ni crear una unidad virtual en este estado por lo que, previamente, hay que parar el servicio:

```

# /etc/init.d/drbd stop

```

Si por el contrario no se muestra ningún error, es entonces el momento de arrancar *drbd* en ambos nodos:

```

# /etc/init.d/drbd start

```

Una vez que *DRBD* se encuentra arrancado en ambos nodos se ha de proceder a establecer la comunicación entre los dos nodos para que pueda dar comienzo la sincronización inicial. Para conectarlos hay que ejecutar el siguiente comando:

```

#drbdadm connect nombre_recurso

```

Es posible ver el estado de la sincronización de las particiones de disco mediante el comando:

```
# cat /proc/drbd
```

Por último, y una vez que se ha determinado la partición virtual en ambos nodos y ambos nodos se han conectado entre sí, es necesario, realizar la sincronización inicial entre los dispositivos de bloque de datos creados. Este paso solo ha de realizarse en un nodo, y solo en el nodo cuyos datos se desean conservar y replicar. Es muy importante prestar especial atención a este punto para no realizar la sincronización inicial en la dirección equivocada. Para ello, ejecutar en el nodo cuyos datos se quieren conservar y replicar (es decir, el nodo que desea marcarse como primario), el siguiente comando:

```
# drbdsetup /dev/drbd0 primary -o
```

Los datos comenzarán a sincronizarse. El tiempo que dura el proceso dependerá de la velocidad de la red, del tamaño de la partición a replicar así como de la velocidad de transferencia indicada en el archivo de configuración de *DRBD*.

Una vez finalizada la sincronización se puede comprobar mediante el comando *cat/proc/drbd* si *DRBD* ya se encuentra preparado. Para ello *cs* ha de ser *Connected*, indicando que ambos nodos están conectados, *ld* ha de ser *UpToDate* indicando que los datos se encuentran sincronizados y actualizados y *st* ha de ser *Primary/Secondary* en un nodo y *Secondary/Primary* en el otro quedando así perfectamente definido el rol *DRBD* de cada uno de los nodos, tal y como se muestra en la figura 20. Finalmente *DRBD* está operativo.

```
root@ast1:/home/usuario# cat /proc/drbd
version: 8.3.7 (api:88/proto:86-91)
srcversion: EE47D8BF18AC166BE219757
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----
    ns:1012024 nr:0 dw:0 dr:1012232 al:0 bm:62 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos
:0
```

Figura 20. Salida mostrada por el comando `cat /proc/drbd` indicando que ambos nodos están conectados y sincronizados

Fuente: El autor

La idea ahora es la de crear un sistema de ficheros en la partición virtual, montarla para poder hacer uso de ella como si se tratara de una partición normal. Es importante destacar que estos últimos pasos sólo se ejecutan en el nodo que fue marcado como nodo primario.

Luego de instalado y configurado DRBD, el siguiente paso es tomar los ficheros de configuración de Asterisk y los directorios donde se almacena la información de los usuarios para colocarlos en la partición virtual que se acaba de montar. Los directorios y ficheros que deben moverse son los siguientes:

```
/etc/asterisk
/var/lib/asterisk
/usr/lib/asterisk
/var/spool/asterisk
/var/log/asterisk
```

Luego, se deben mover los directorios y ficheros de asterisk hacia la partición virtual por medio de las siguientes instrucciones:

```
# cd /mnt/local/replica/asterisk
asterisk# cp -r /etc/asterisk etc/asterisk
asterisk# cp -r /var/lib/asterisk var/lib/asterisk
asterisk# cp -r /usr/lib/asterisk usr/lib/asterisk
asterisk# cp -r /var/spool/asterisk var/spool/asterisk
asterisk# cp -r /var/log/asterisk var/log/asterisk
```

Posteriormente, se crean los enlaces desde estas nuevas ubicaciones hacia sus ubicaciones originales de manera que no se vea alterado el funcionamiento de Asterisk:

```
# ln -s /mnt/local/replica/asterisk/etc/asterisk /etc/asterisk
# ln -s /mnt/local/replica/asterisk/var/lib/asterisk /var/lib/asterisk
# ln -s /mnt/local/replica/asterisk/usr/lib/asterisk /usr/lib/asterisk
# ln -s /mnt/local/replica/asterisk/var/spool/asterisk /var/spool/asterisk
# ln -s /mnt/local/replica/asterisk/var/log/asterisk /var/log/asterisk
```

Este conjunto de instrucciones de ejecutarse en el nodo primario y luego en el resto de los nodos del clúster. En estos últimos los enlaces quedarán rotos hasta el momento en que tengan que tomar el control

Soporte para Alta Disponibilidad en Servicios

De la misma manera como se hizo con el software de alta disponibilidad de datos, se procedió con el software de alta disponibilidad en servicios seleccionando el Proyecto LINUX-HA al comienzo de este capítulo. El proyecto Linux-HA tiene como objetivo proporcionar una solución de alta disponibilidad para Linux que promueva la fiabilidad y disponibilidad de sistemas a través de su comunidad de desarrolladores.

La arquitectura presentada en la actualidad por Linux-HA se muestra en la figura 21, cuyos componentes serán descritos detalladamente según Villar, E. (2010):

- 1. STONITH:** renombrado como ClusterGlue, es un elemento imprescindible en la arquitectura de alta disponibilidad de Linux-HA ya que permite la interoperabilidad entre sus diferentes elementos.
- 2. Resource Agents:** son los agentes de recurso, un extenso conjunto de scripts para la gestión y acceso a los diferentes recursos y servicios que son ofrecidos y mantenidos por el *clúster*.

3. **Heartbeat:** es ahora solamente la capa que comunica los nodos del clúster, permitiendo el intercambio de mensajes para determinar la presencia o la ausencia de pares de procesos entre los nodos. Para alcanzar el mayor rendimiento posible Heartbeat debe ser usado en combinación con Pacemaker, encargado de llevar a cabo las tareas de arranque y parada de los servicios a los que se quiere dotar de alta disponibilidad.
4. **Pacemaker:** es el nuevo nombre asignado al módulo CRM (gestor de los recursos del clúster), encargado de arrancar los recursos o servicios, detenerlos, configurarlos, monitorizarlos, establecer la configuración global del clúster, entre otros. A través de pacemaker es posible configurar de forma directa (antes había que hacerlo editando ficheros XML) el clúster.

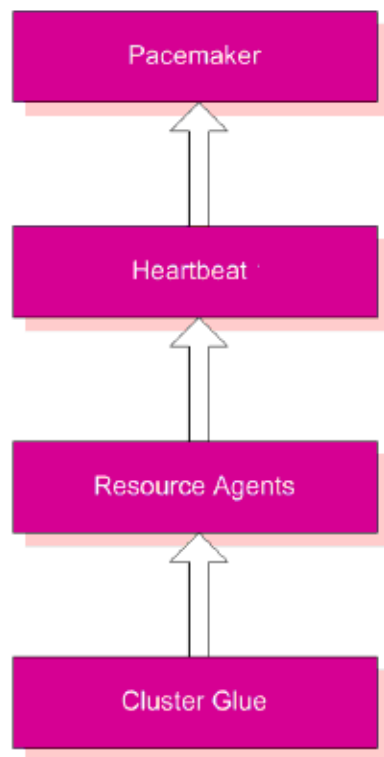


Figura 21. Arquitectura del Proyecto LINUX-HA

Fuente: Villar, E. (2010)

Instalación de LINUX-HA:

A continuación se describe el proceso de instalación de cada uno de los elementos que componen LINUX-HA, siguiendo la propuesta de Villar, E., (2009), la cual fue adaptada según los requerimientos del caso del estudio del presente trabajo

Inicialmente se deben instalar ciertas librerías que son imprescindibles para que *Linux-HA* y sus componentes puedan operar de forma correcta:

```
# apt-get install xsltproc docbook-xml docbook-xsl
# apt-get install build-essential libglib2.0-dev
  pkg-config
```

Luego se procede a descargar e instalar la última versión estable para cada uno de los componentes del proyecto *Linux-HA* en el siguiente orden:

1. *Instalación de ClusterGlue*: se deben instalar las siguientes para *ClusterGlue*:

```
# apt-get install autoconf libtool libglib2.0-dev flex bison libxml2-dev libbz2-dev
  uuid-dev automake
```

Luego de instaladas las dependencias anteriores se descarga la última versión estable de *ClusterGlue* desde los repositorios *mercurial* de *Linux-HA* mediante el comando *wget*:

```
# wget http://hg.linux-ha.org/glue/archive/glue 1.0.5.tar.bz2
```

Con el paquete descargado, con el uso del comando *tar* se descomprime y se ingresa al directorio de la descompresión:

```
# tar xjvf glue-1.0.5.tar.bz2
# cd Reusable-Cluster-Components-glue-1.0.5/
```

Después, desde el directorio *Reusable-Cluster-Components-glue-1.0.5* se ejecutan los *scripts* de preparación de la instalación *autogen.sh* y *configure*:

```
Reusable-Cluster-Components-glue-1.0.5# ./autogen.sh
```

```
Reusable-Cluster-Components-glue-1.0.5# ./configure
```

Finalmente, se procede a compilar el código fuente y realizar la instalación ejecutando *make* y *make install*, respectivamente:

```
Reusable-Cluster-Components-glue-1.0.5# make  
Reusable-Cluster-Components-glue-1.0.5# make install
```

Con *ClusterGlue* instalado, se continúa con el siguiente nivel en la arquitectura *Linux-HA* instalando *ResourceAgents*.

2. *Instalación de ResourceAgents*: Se descarga el paquete con las fuentes del agente de recursos y luego se descomprime:

```
# wget http://hg.linux-ha.org/agents/archive/agents-1.0.3.tar.bz2  
# tar xjvf agents-1.0.3.tar.bz2
```

Después se ubica al directorio obtenido en la descompresión para ejecutar los scripts *autogen.sh* y *configure*, y compilar y realizar la instalación mediante los comandos *make* y *make install*:

```
# cd Cluster-Resource-Agents-agents-1.0.3/  
Cluster-Resource-Agents-agents-1.0.3# ./autogen.sh  
Cluster-Resource-Agents-agents-1.0.3# ./configure  
Cluster-Resource-Agents-agents-1.0.3# make  
Cluster-Resource-Agents-agents-1.0.3# make install
```

Una vez instalado *ResourceAgents*, se procede a instalar *Heartbeat* 3.

3. *Instalación de Heartbeat*: Antes de instalar *Heartbeat*, se deben cumplir algunas dependencias de utilidades y librerías necesarias para configurar y llevar a cabo el

proceso de instalación, para lo cual se utiliza el comando apt-get install como hasta ahora:

```
# apt-get install libbz2-dev perl-base libperl-dev libltdl3-dev libxml2-dev libnet1-dev libglib2.0-dev gawk zlib1g zlib1g-dev
```

Luego de instalar las dependencias, se debe descargar el paquete estable de Heartbeat desde los repositorios mercurial oficiales de Linux-HA usando el comando wget:

```
# wget http://hg.linux-ha.org/heartbeat\_STABLE\_3\_0/archive/STABLE-3.0.3.tar.bz2
```

Descomprimir el paquete descargado usando tar, para luego ingresar en el directorio creado durante la descompresión, que contiene el código fuente de Heartbeat 3:

```
# tar xjvf STABLE-3.0.3.tar.bz2
# cd Heartbeat-3-0-STABLE-3.0.3/
```

Una vez ubicado el directorio, se ejecuta el comando bootstrap:

```
Heartbeat-3-0-STABLE-3.0.3# ./bootstrap
```

Para el proceso de instalación y configuración de Heartbeat se recomienda ejecutar el script ConfigureMe con las opciones configure, make y make install, respectivamente. Además, a cada una de las ejecuciones se la agrega la opción “ – enable-fatalwarnings=no” para evitar que avisos impidan la culminación adecuada de la instalación:

```
Heartbeat-3-0-STABLE-3.0.3# ./ConfigureMe configure --enable-fatalwarnings=no
Heartbeat-3-0-STABLE-3.0.3# ./ConfigureMe make --enable-fatalwarnings=no
Heartbeat-3-0-STABLE-3.0.3# ./ConfigureMe install --enable-fatalwarnings=no
```

En este punto ya se dispone de Heartbeat 3 instalado, pero debe adecuarse el sistema para que Heartbeat trabaje. Se debe crear primero un grupo de usuarios denominado haclient y un usuario hacluster dentro de este grupo:

```
# addgroup haclient  
# adduser hacluster -gid 1000
```

El usuario es añadido al grupo en el momento de su creación especificando el identificador de grupo gid que se obtiene del fichero /etc/group. Después, se hace a hacluster y a haclient propietarios de los ficheros de Heartbeat con el comando chown:

```
# chown -R hacluster:haclient /var/run/heartbeat  
# chown -R hacluster:haclient /var/lib/heartbeat  
# chown -R hacluster:haclient /usr/lib/heartbeat  
# chown -R hacluster:haclient /usr/var/run/heartbeat
```

Finalmente, antes de poner en marcha el servicio Heartbeat es necesario copiar los siguientes ficheros desde el subdirectorio doc del directorio principal de las fuentes descargadas en el directorio /etc/ha.d/:

```
Heartbeat-3-0-STABLE-3.0.3# cp doc/authkeys /etc/ha.d/  
Heartbeat-3-0-STABLE-3.0.3# cp doc/ha.cf /etc/ha.d/  
Heartbeat-3-0-STABLE-3.0.3# cp doc/haresources /etc/ha.d/
```

Es necesario también cambiar los permisos de acceso al fichero authkeys (el primero de los copiados en el código anterior) para que solo los propietarios tengan acceso de lectura y escritura al mismo:

```
Heartbeat-3-0-STABLE-3.0.3# chmod 600 /etc/ha.d/authkeys
```

Hasta aquí la instalación de Heartbeat. El siguiente paso es su configuración. Se debe establecer en cada nodo del clúster un nombre único por el cual identificarlo, editando en el caso de Debian 6.0 el fichero /etc/hostname.

Finalmente, se debe editar el fichero `/etc/hosts` en cada nodo del clúster, incluyendo una línea por nodo de la siguiente forma:

```
<direccion_ip> <nombre del nodo>
```

Así, cada nodo puede comunicarse con el resto de los nodos.

4. Instalación de Pacemaker: Se debe instalar la librería *libxslt-dev*:

```
# apt-get install libxslt-dev
```

Se descargan en primer lugar las fuentes desde los repositorios *mercurial* de *Cluster Labs* en la Web oficial del proyecto *Pacemaker* (<http://www.clusterlabs.org/>):

```
# wget http://hg.clusterlabs.org/pacemaker/stable-1.0/archive/Pacemaker-1.0.8.tar.bz2
```

Las fuentes de la utilidad se descomprimen con *tar* y se ubica al directorio generado:

```
# tar xjvf Pacemaker-1.0.8.tar.bz2
# cd Pacemaker-1-0-Pacemaker-1.0.8/
```

Luego se procede a ejecutar los *scripts autogen-sh* y *configure*, éste último con la opción “*--with-heartbeat*” para especificar que *Heartbeat* es el que actúa como capa de comunicación/mensajes entre los nodos del clúster:

```
Pacemaker-1-0-Pacemaker-1.0.8# ./autogen.sh
Pacemaker-1-0-Pacemaker-1.0.8# ./configure --with-heartbeat
```

Ejecutamos *make* y *make install* para finalizar compilando e instalando *Pacemaker*:

```
Pacemaker-1-0-Pacemaker-1.0.8# make
```

Pacemaker-1-0-Pacemaker-1.0.8# make install

Luego de completada la instalación de todos los componentes de LINUX-HA, en cada uno de los nodos que formarán parte del clúster, se presenta a continuación la configuración en la cual se basa la propuesta de alta disponibilidad en servicios para el Sistema actual de Telefonía IP de la sede administrativa principal del IPASME.

Configuración de todos los nodos del Cluster:

- 1) En el fichero /etc/ha.d/authkeys se configuran las claves de las que hacen uso los nodos del clúster para autenticarse, habiendo tres esquemas diferentes posibles para ser usados: crc, md5 o sha1. Es decir, en este fichero se configuran los aspectos de seguridad del clúster. El uso de uno u otro modo de autenticación depende de las exigencias de seguridad y rendimiento que se establezcan: a). **El método crc** es recomendable cuando los nodos del clúster se comunican a través de una red segura cuando se necesita disponer de la máxima cantidad de recursos; b). **La autenticación mediante sha1** que requiere una mayor cantidad de recursos de CPU en su aplicación al mismo tiempo que aporta una gran seguridad; c). **md5** es una solución intermedia a los dos métodos anteriores.

Para el sistema de estudio, se plantea que los servidores estén directamente conectados ó a través de un switch Gigabit Ethernet , por lo que no son necesarias grandes medidas de seguridad y sería suficiente con aplicar el método crc. Para ello sólo es necesario incluir las siguientes líneas en fichero authkeys:

auth 1

1 crc

- 2) La configuración principal del clúster es la que se incluye en el fichero ha.cf, ubicado en el directorio /etc/ha.d/. El fichero contiene todas las características configurables para el clúster, las cuales pueden ser habilitadas o deshabilitadas en

función de la configuración requerida; para ello basta con descomentar o comentar la línea correspondiente, respectivamente. De manera general, se puede decir que algunos de los aspectos configurables son los ficheros que almacenan información sobre la ejecución de los servicios como log, los parámetros que rigen la monitorización de recursos y nodos, puertos e interfaces de escucha para Heartbeat o los nodos que conforman el clúster. Para la configuración del cluster de la propuesta de estudio, los parámetros configurados dentro de este archivo son:

- a) *debugfile*. Permite especificar la ubicación del fichero que almacena los mensajes generados por Heartbeat en modo depuración. Para el diseño propuesto este fichero puede ser ubicado en el directorio `/var/log/ha-debug`
- b) *logfile*. Al igual que la opción anterior, permite generar el resto de mensajes generados como log. Para el diseño propuesto este fichero puede ser ubicado en el directorio `/var/log/ha-log`
- c) *keepalive*. Es el tiempo, por defecto en segundos (aunque es posible especificarlo en milisegundos escribiendo ms) que transcurre entre la emisión de dos pings hacia cada nodo del clúster para su monitorización. Para este diseño se estableció en 2 segundos
- d) *deadtime*. Permite especificar el tiempo en segundos que deben transcurrir para asumir que un nodo no se encuentra disponible o que está muerto. Este parámetro para el diseño propuesto se asumió en 20 segundos
- e) *warnetime*. En esta opción se especifica el tiempo en segundos que transcurre antes de indicar la advertencia de último intento de ping en la monitorización. Si es habitual alcanzar el último intento de ping entonces quiere decir que se esta próximo a considerar el nodo como muerto. El tiempo establecido para la propuesta fue de 10 segundos.
- f) *initdead*. Es el tiempo en segundos que debe transcurrir de manera obligatoria antes de la puesta en marcha de Heartbeat. Si, por el motivo que fuese un nodo no logra iniciarse antes de que transcurra este tiempo es considerado

como un nodo muerto. La configuración de este parámetro se ajustó a 20 segundos para el diseño propuesto.

- g) *udpport*. Permite especificar el número de puerto que utiliza Heartbeat para las comunicaciones de tipo unicast y broadcast. Este puerto es el mismo tanto para el envío de pings de monitorización al resto de nodos como para su recepción. El puerto que usa Heartbeat en el diseño propuesto es el 694
- h) *bcast*. Interfaz de red que usa Heartbeat para el envío de los pings. El valor de este parámetro depende estrictamente de la interfaz que se vaya a tomar en cada servidor para las comunicaciones del cluster.
- i) *autofailback*. Esta opción sirve para especificar el comportamiento de Heartbeat cuando a causa de un fallo en el nodo activo, los servicios son migrados a otro nodo y luego solventen los problemas en el nodo primario. Hay dos opciones posibles: que una vez que el nodo que falló se recupera los recursos vuelvan a él, o que permanezcan en el nuevo nodo activo. Para la primera opción el valor de este argumento es “*on*”, mientras que para la segunda es *off*. Se recomienda colocar este valor en “*off*” para evitar posibles problemas de sincronización entre los nodos.
- j) *node*. Especifica los nodos que forman parte del clúster. Para cada nodo se debe escribir el nombre que se obtiene al ejecutar en él el comando `hostname` o `uname -n`.
- k) *crm*. Define el uso o no del administrador de recursos del clúster CRM (Cluster Resource Manager). Se coloca “*yes*” para permitirlo ó “*no*” para no hacerlo. Para el diseño propuesto se colocó el valor “*yes*”.

Configuración de uno de los nodos del Cluster:

La configuración del clúster continúa ahora desde sólo uno de los nodos ya que los cambios que se originen se replican y se distribuyen al resto de los nodos. Por medio de la configuración que se describe a continuación, se define la

funcionalidad del clúster: qué servicios van a ser ejecutados y en qué nodos, cómo van a ser monitorizados, qué tipo de comunicaciones se establecen entre los nodos, cuando se produce la migración de los servicios o grupos de servicios. Para configurar todos estos aspectos se utiliza la aplicación Pacemaker.

Como se dijo antes, Pacemaker es el administrador de recursos del clúster (Cluster Resource Manager o CRM) que se usa de manera conjunta con Heartbeat en un sistema de alta disponibilidad Linux-HA. Con Pacemaker es posible editar el fichero `cib.xml` (por sus siglas en inglés Cluster Information Base), ubicado en el directorio `/usr/var/lib/heartbeat/crm/`, el cual recoge toda la funcionalidad del clúster y cómo éste se encuentra configurado.

De manera general, el diseño de alta disponibilidad de servicios propuesto para el Sistema de Telefonía IP de la sede administrativa principal del IPASME, se desarrolla bajo las siguientes directrices: a). Se tiene un clúster formado por dos nodos; b). Se debe definir un recurso para la ejecución de Asterisk el cual debe estar dotado de Alta disponibilidad, siendo ejecutado solamente en uno de los nodos del cluster; c). Se debe disponer de un recurso de tipo IP virtual que permita alcanzar al servicio Asterisk sea cual sea su ubicación; d). El recurso definido para Asterisk y el recurso de la IP virtual deben conformar un grupo de recursos (ya que el recurso Asterisk hace uso del recurso de IP virtual, es indispensable que ambos recursos siempre estén corriendo en el mismo nodo); e). Se deben definir las restricciones de localización, colocación y orden para los recursos del clúster.

1) *Configuración del fichero `cib.xml`: iniciar el servicio `Heartbeat` en cada uno de los nodos del clúster, ejecutando el script disponible tras su instalación en el directorio `/etc/init.d`:*

```
nodo1:~# /etc/init.d/heartbeat start
```

```
nodo2:~# /etc/init.d/heartbeat start
```

Luego que Heartbeat se encuentra en ejecución, iniciar la utilidad CRM (Cluster Resource Manager) de Pacemaker, (para configurar todos los detalles de la funcionalidad para el clúster de alta disponibilidad de servicios) en uno de los nodos del clúster. Para acceder al intérprete `crm` se ejecuta el siguiente comando:

```
nodo1:~# crm
```

```
crm(live)#
```

Al escribir *help* en el primer nivel se pueden observar los diferentes menús de configuración y comandos disponibles para comenzar, como se aprecia en la figura 22.

De los distintos menús disponibles en `crm` las opciones que se utilizarán durante la configuración del diseño propuesto son:

- a) *Cib*: permite editar y configurar el fichero `cib.xml`. `Crm` guarda en diferentes ficheros denominados `cib shadow` las distintas configuraciones que se vayan realizando. Es decir, se puede mantener un fichero por cada servicio que ofrece el clúster y aplicarlo cuando sea conveniente. Desde `cib` se pueden crear nuevos ficheros `shadow`, editar los existentes o eliminarlos, elegir el que se quiera usar, distribuir los cambios realizados al resto de nodos del clúster, entre otros.

```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
root@ast1:/home/usuario# crm
crm(live)# help

This is the CRM command line interface program.

Available commands:

      cib          manage shadow CIBs
      resource     resources management
      node         nodes management
      options      user preferences
      configure    CRM cluster configuration
      ra           resource agents information center
      status       show cluster status
      quit,bye,exit exit the program
      help         show help
      end,cd,up    go back one level
```

Figura 22. Menús disponibles para trabajar con crm
Fuente: el autor (2011)

- b) *Resource*: Escribiendo resource se accede a la administración de los recursos del clúster. Desde allí es posible conocer el estado de los recursos, listarlos, cambiar su estado (iniciarlos, detenerlos, reiniciarlos...), migrarlos a otro nodo del clúster, editar los parámetros de los mismos así como sus metadatos, ver la cuenta de fallos asociada.
- c) *Node*: permite el acceso a la administración de los nodos que forman parte del clúster. Dispone de comandos para mostrar o listar los nodos del clúster, cambiar el estado de un nodo (online/offline), apagarlo, eliminarlo de la configuración ó también conocer el estado de sus atributos y editarlos si es necesario.
- d) *Configure*: Permite la configuración de los ficheros cib shadow. A través de él se puede realizar la administración de nodos y recursos en el clúster, la administración de grupos de recursos, la definición de las restricciones de localización/orden para los recursos, o ver propiedades generales del clúster.

- e) Status: Permite conocer el estado del clúster. Muestra si los nodos se encuentran actualmente online, qué servicios se encuentran ejecutándose y en qué nodos, cuáles fueron los últimos fallos detectados.

Volviendo a la configuración del fichero `cib.xml` en el diseño propuesto, como primer paso dentro de `crm` se debe crear el fichero `cib shadow` en el que se va almacenar la configuración del clúster. Esto se logra ejecutando el comando `new` en el nivel `cib` y escribiendo el nombre que se le va asignar al fichero, por ejemplo `HA_Asterisk`:

```
crm(live)# cib new HA_Asterisk
crm(HA_Asterisk)# cib use HA_Asterisk
```

Luego, se crean los recursos individuales sobre los que va a operar Heartbeat (dirección IP virtual y Asterisk). Ahora se debe descender un nivel y operar en `configure`:

```
crm(HA_Asterisk)# configure
```

Con `crm` los recursos son creados mediante el comando `primitive`. `Primitive` acepta como parámetros el nombre para el recurso, el agente de recurso que lo administra, los parámetros que éste precisa y las diferentes operaciones que debe soportar. Además, se puede definir el valor para algunas de las características del recurso, como por ejemplo: su estado inicial. Para definir el recurso IP virtual se procede como sigue:

```
crm(HA_Asterisk)configure# primitive failover-ip ocf:heartbeat:IPaddr params
ip="150.214.153.235" op monitor interval="10s" meta target-role="Started"
```

Detallando la instrucción anterior, se tiene que: Se asigna el nombre `failover-ip` al recurso, se indica que el agente del recurso fue creado siguiendo el estándar `ocf:heartbeat` y se llama `IPaddr` (este agente de recurso es proporcionado por el

propio Heartbeat). Como parámetro al agente del recurso se le añade la dirección de red que será establecida como IP virtual: 150.214.153.235, por ejemplo.

Un recurso debe poder soportar cuatro operaciones diferentes: *start*, *stop*, *status* y *monitor*. Si no se especifican valores para estas operaciones, Heartbeat toma los valores por defecto de los que dispone. Para el manejo de los recursos en el diseño propuesto se concretó el intervalo de tiempo entre la ejecución de dos operaciones *monitor* en 10 segundos. Al final, mediante la etiqueta *meta* se le indica a Heartbeat que el estado inicial para el recurso debe ser *Started* (iniciado).

De forma similar se debe definir el **recurso para el servicio Asterisk**, pero con una importante salvedad: *Heartbeat* no dispone de un *script* que actúe como *agente de recurso* para *Asterisk* de manera predeterminada. En este caso, se debe utilizar como agente de recurso, el *script asterisk* de inicio y parada del servicio instalado con *Asterisk*, y que se encuentra en el directorio */etc/init.d*

Para poder hacer uso de este *script* y las funciones *monitor* y *status* que contiene, es necesario crear un grupo de usuarios *asterisk* e instalar la utilidad para el rastreado de redes y servicios de red *nmap*:

```
# groupadd asterisk  
# apt-get install nmap
```

También es necesario eliminar todas las referencias para iniciar el servicio *Asterisk* de forma automática cada vez que el sistema es puesto en marcha (ejecutando el *script asterisk*), ya que a partir de ahora va a ser *Heartbeat* el encargado de iniciar y detener este servicio. Para ello se ejecuta el siguiente comando:

```
nodo1:~# update-rc.d -f asterisk remove
```

La definición del recurso *failover-asterisk* queda de la siguiente forma:
crm(HA_Asterisk)configure# primitive failover-asterisk lsb:asterisk op monitor interval="15s" meta target-role="Started"

A diferencia de los parámetros vistos para el recurso *IP virtual*, ahora el agente del recurso es del tipo *lsb* y la operación *monitor* es llevada a cabo cada 15 segundos. Es decir, *Heartbeat* comprueba cada 15 segundos que *Asterisk* está siendo ejecutado con normalidad y al final, mediante la etiqueta *meta* se le indica a *Heartbeat* que el estado inicial para el recurso debe ser *Started* (iniciado).

Con los dos recursos individuales creados, a continuación se procede a crear el grupo de recursos al que van a pertenecer. Para crear un grupo de recursos se debe utilizar en el nivel *configure* la cláusula ***group***, seguida del nombre que se le va a asignar al grupo así como los nombres de los recursos que pertenecen a él, en el orden en el que estos deben ser iniciados/detenidos. De esta manera se está creando la restricción de orden en la que la *IP virtual* debe estar disponible antes de que el servicio *Asterisk* sea iniciado:

```
crm(HA_Asterisk)configure# group asterisk-group failover-ip failover-asterisk
```

Creando el grupo de recursos *asterisk-group* también se establece la restricción de colocación necesaria para el escenario del diseño propuesto: los recursos *IP virtual* y *asterisk* deben ejecutarse siempre en el mismo nodo. Antes de proseguir configurando la monitorización del grupo de recursos por parte de *Heartbeat* se deben establecer los valores para dos de las propiedades globales del ***clúster***: ***no-quorum-policy*** y ***stonith-enabled***. Con la primera se establece la política *quorum* para el *clúster*, es decir, cómo debe actuar en el caso en que el problema *SplitBrain* sea detectado. Para el diseño propuesto esta política debe configurarse *ignore* (ignorar). Mediante la segunda se habilita/deshabilita el que un dispositivo adicional al *clúster* (*dispositivo stonith*) pueda apagar o reiniciar un nodo que ha experimentado dificultades; como no es el objetivo del proyecto establecer una configuración excesivamente compleja, se deshabilita esta propiedad (con el valor *false*). Para modificar los valores de las propiedades del *clúster* se usa la palabra clave ***property***, también en el nivel *configure*:

```
crm(HA_Asterisk)configure# property no-quorum policy="ignore"
```

```
crm(HA_Asterisk)configure# property stonith enabled="false"
```

Durante la configuración del comportamiento del *clúster* mediante *crm* siempre existe la posibilidad de verificar la configuración y que no existen errores en los procedimientos y sintaxis, por medio del comando **verify**:

```
crm(HA_Asterisk)configure# verify
```

Si la ejecución de *verify* no devuelve nada, ningún problema ha sido detectado. También es útil observar cada cierto tiempo el contenido del fichero *cib shadow* que se está creando ejecutando el comando **show**:

```
crm(HA_Asterisk)configure# show
```

Ahora debe añadirse la restricción de localización para el grupo de recursos *Asterisk-group*. Continuando en el nivel *configure*, se usa la palabra clave **location** utilizando la sintaxis siguiente:

```
crm(HA_Asterisk)configure# location run_asterisk-group asteriskgroup rule 100:  
#uname eq ast1 rule 90: #uname eq ast2
```

En la instrucción anterior, en primer lugar se escribe un identificador (nombre que se le da a la restricción), por ejemplo *run_asterisk_group*. Después se debe especificar el nombre del recurso para el que se establece la restricción, en este caso el nombre del grupo de recursos (*asterisk-group*). Para finalizar se debe escribir el nombre del nodo preferido para la ejecución del recurso junto con una puntuación o detallar una regla para determinar esta preferencia (las reglas son creadas con una puntuación y una expresión a evaluar). Si la preferencia es resuelta favorablemente entonces la puntuación es incrementada en el valor indicado.

En este último aspecto, se establecen los valores para las propiedades: **default-resource-stickiness** y **default-resourcefailure-stickiness**, las cuales establecen la

adherencia por defecto de los recursos para ser ejecutados en el nodo actual o por el contrario ser migrados a otro nodo que ofrece unas condiciones mejores. Se debe asignar una puntuación positiva en *default-resource-stickiness* para premiar la correcta ejecución del grupo en un nodo y una puntuación negativa en *defaultresource-failure-stickiness* para penalizar la ejecución del grupo en un nodo si han sido detectado fallos. Esto se configura utilizando de nuevo en comando *property*:

```
crm(HA_Asterisk)configure# property default-resource-stickiness=20
crm(HA_Asterisk)configure# property default-resource-failurestickiness=-20
```

Aunque estos valores son *por defecto* para cualquier recurso definido en el fichero del *clúster*, se agregan estas propiedades con sus valores respectivos de forma local en el grupo de recursos *asterisk-group*. Para ello es necesario salir del nivel *configure* ejecutando **end**. Antes, es necesario guardar los cambios realizados hasta ahora en la configuración ejecutando el comando **commit**:

```
crm(HA_Asterisk)configure# commit
crm(HA_Asterisk)configure# end
```

Es necesario también comunicar los cambios realizados al resto de nodos del *clúster*. Esto se logra con la ejecución del comando **commit** también, pero en el nivel **cib** y escribiendo además el nombre del fichero *cib shadow* que se quiere distribuir:

```
crm(HA_Asterisk)# cib commit HA_Asterisk
```

Finalmente, se establecen los valores para las propiedades **resource-stickiness** y **resource-failure-stickiness** para el grupo de recursos cambiando al nivel **resource** y haciendo uso del comando **meta** de la siguiente forma:

```
crm(HA_Asterisk)resource# meta asterisk-group set
resourcestickiness 20
```

```
crm(HA_Asterisk)resource# meta asterisk-group set
resource-failurestickiness -20
```

Como último paso se deben establecer los mecanismos por los cuales *Heartbeat* va a determinar si es necesario migrar los servicios del nodo *activo* al otro o no. Para ello es necesaria la definición de las operaciones de monitorización de cada uno de los recursos individuales que forman el grupo *asterisk-group* (*failover-ip* y *failover-asterisk*) y hacer uso de la propiedad ***migration-threshold***.

Tanto para *failover-ip* como para *failover-asterisk* se debe establecer el valor de la propiedad ***timeout*** para la operación ***monitor***, ya que permite fijar el *timeout* o tiempo máximo permisible para la ejecución de la operación. Así *Heartbeat*, una vez que transcurre este tiempo sin haber culminado la operación correctamente, determina que algo debe ir mal en la ejecución/disponibilidad del recurso. Los *timeouts* para las operaciones deben ser al menos de la misma duración que las recomendaciones disponibles en *meta-data* (si se establece un valor no recomendado se recibe un alerta), aunque eso no quiere decir que para un recurso determinado sea suficiente. Un error común en la configuración de este tipo de *clústeres* es establecer valores demasiado pequeños para los *timeouts* de las operaciones.

La forma más cómoda de cambiar la configuración de la operación *monitor* una vez que ésta ya sido definida anteriormente es la de editar el fichero *cib shadow* directamente de forma manual. Para ello, en el nivel ***configure*** se ejecuta el comando ***edit***. Entonces el fichero es abierto con el editor seleccionado por defecto (puede ser cambiado a través del nivel ***options***) y se ajusta:

timeout="20s" para las operaciones *monitor* en los dos recursos del *clúster*,

Con los *timeouts* establecidos para la monitorización de los dos recursos individualmente, se pasa a cambiar la propiedad ***migration-threshold*** para el grupo de recursos. Esta propiedad permite fijar el número de fallos o errores que

pueden tener lugar para el recurso o grupo de recursos en el que se aplica para el nodo *actual*. Una vez alcanzado este límite, el recurso o grupo de recursos es migrado a otro nodo. La utilidad de esta propiedad es grande, ya que permite implementar un mecanismo para la migración de los servicios de una forma muy sencilla y eficiente. Para el diseño propuesto se recomienda ajustar este umbral en 1, es decir, con tan solo detectar un fallo para el grupo de recursos *asterisk-group* éste es migrado a otro nodo para continuar con su ejecución:

```
crm(HA_Asterisk)resource# meta asterisk-group set migrationthreshold 1
```

Junto a la propiedad *migration-threshold* es necesario introducir y conocer dos procedimientos que permiten en todo momento comprobar el número de fallos registrados por *Heartbeat* para un recurso o grupo de recursos y *limpiar* o *resetear* esta cuenta. Para conocer el valor de esta cuenta se hace uso del comando *failcount* en el nivel *resource*, ya que su valor es referente a un recurso/grupo de recursos concreto y para un determinado nodo.

Por ejemplo para conocer el número de fallos experimentados por el recurso *failover-asterisk* en el nodo *ast1* se deben ejecutar las instrucciones:

```
crm(live)# resource
crm(live)resource# failcount failover-asterisk show ast1
```

Para *limpiar* el valor de la cuenta de fallos se hace uso de esta misma sintaxis y la operación *delete*:

```
crm(live)resource# failcount failover-asterisk delete ast1
```

En este punto se ha completado la configuración de los recursos y el grupo de recursos y es momento de guardar los cambios mediante la siguiente instrucción:

```
crm(HA_Asterisk)configure# commit
crm(HA_Asterisk)configure# end
```

Luego de salir del nivel *configure*, como último paso se deben comunicar los cambios en el fichero *cib shadow* al otro nodo del *clúster*, ejecutando el comando *commit* también pero en el nivel *cib*:

```
crm(HA_Asterisk)# cib commit HA_Asterisk
```

Para salir de *crm* se puede escribir *bye*:

```
crm(HA_Asterisk)# bye
```

A partir de ahora basta con iniciar *Heartbeat* en todos los nodos del *clúster* para poner en marcha el sistema de alta disponibilidad propuesto. Es importante recordar que el servicio debe iniciarse primero en el nodo que se configuró como nodo principal y luego en el resto de los nodos del *clúster*.

Evaluación del diseño propuesto de Alta Disponibilidad

Una vez presentado teóricamente el diseño propuesto donde se detalló cuales son los elementos de alta disponibilidad que pueden incorporarse al Sistema actual de Telefonía IP de la sede administrativa principal del IPASME, se construyó un escenario de pruebas con la finalidad de evaluar la operatividad de dicho diseño.

Para la construcción de este escenario de pruebas se utilizaron técnicas de virtualización con base a la revisión teórica hecha de este tema en el capítulo II. LA arquitectura del escenario de pruebas configurado se muestra en la figura 23

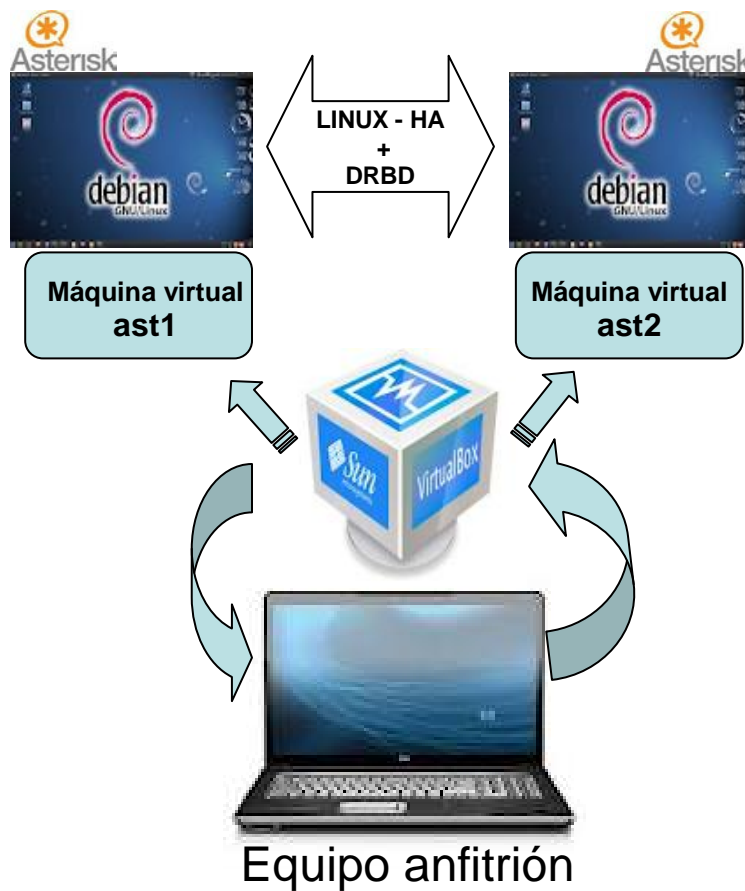


Figura 23. Arquitectura del escenario de pruebas
Fuente: el autor (2011)

Se dispone de un equipo anfitrión donde se instaló la herramienta de virtualización Sun VirtualBox, versión 3.1.6. Sobre esta herramienta se crearon dos máquinas virtuales inicialmente llamadas debian1 y debian2 respectivamente. La configuración del hardware con el que se crearon estas máquinas se muestra en las figuras 24 y 25.

Como Sistema Operativo, ambas máquinas virtuales disponen de Debian 6.0. Antes de comenzar la configuración de la propuesta, es importante que el sistema operativo este actualizado e instalar el software Asterisk en ambas máquinas.

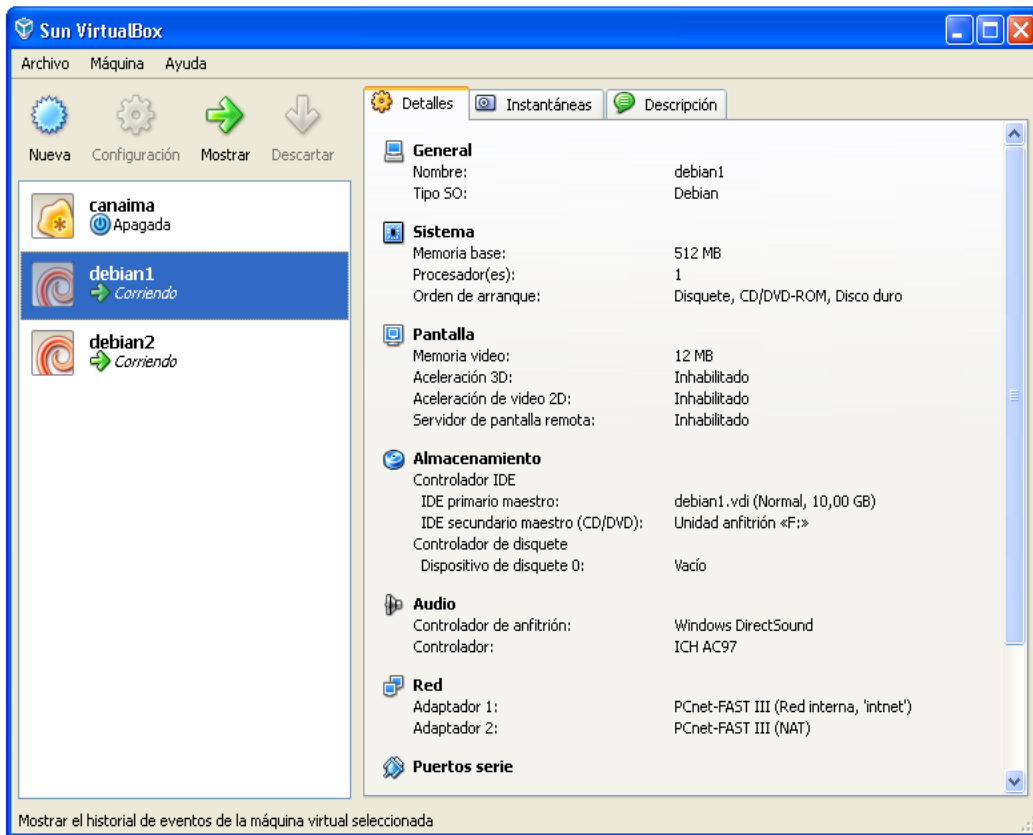


Figura 24. Configuración de hardware de la máquina virtual debian1
Fuente: el autor (2011)

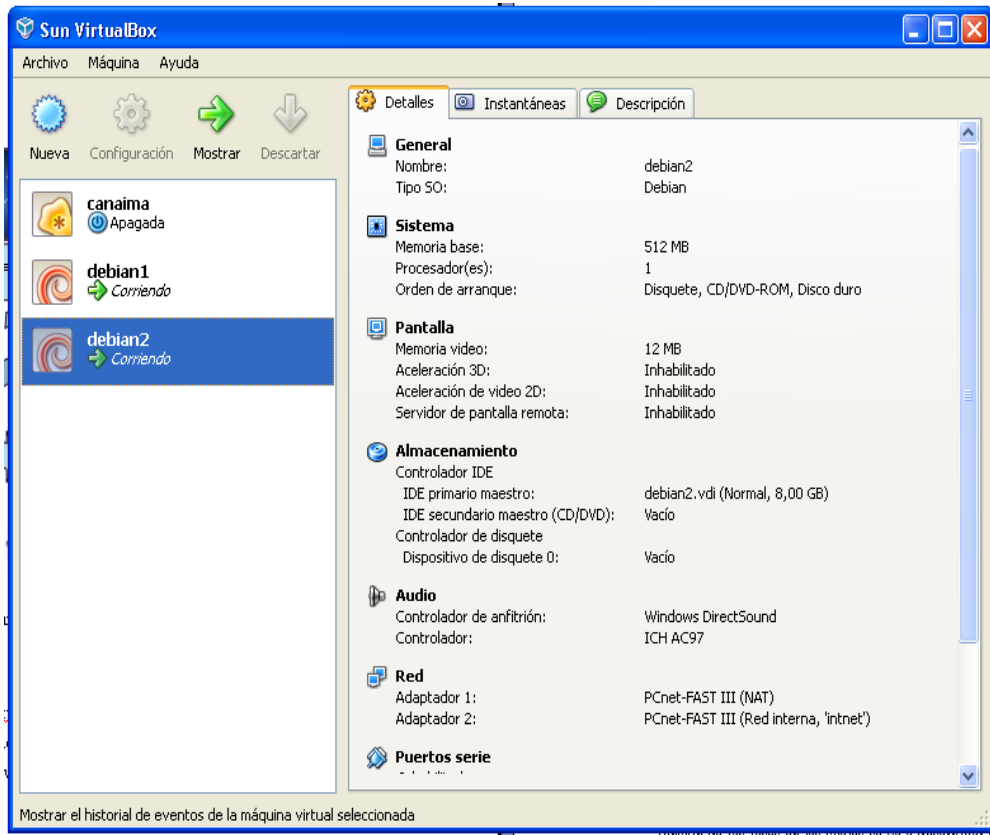
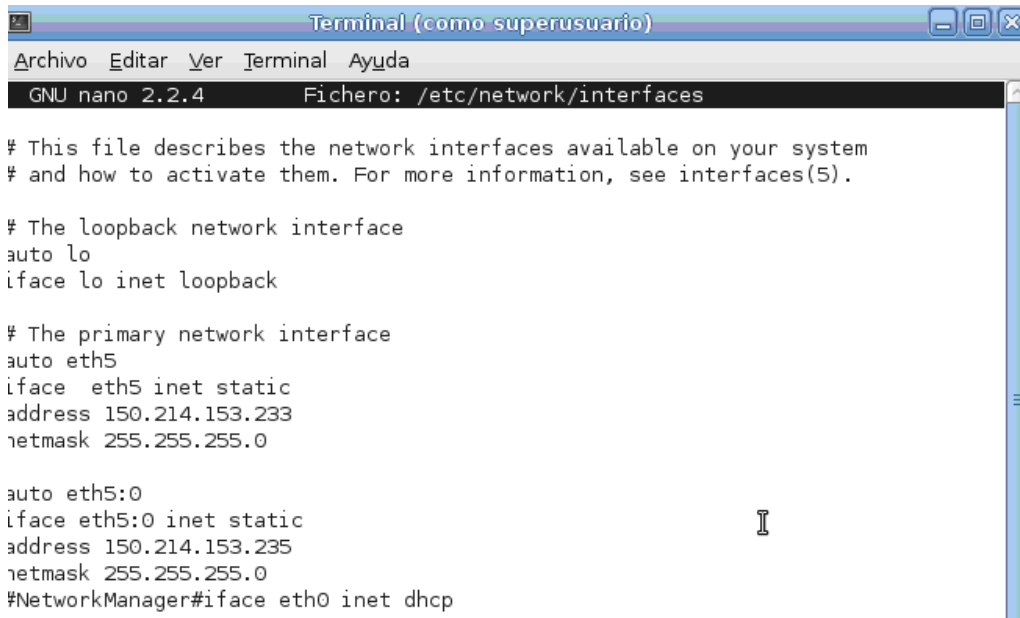


Figura 25. Configuración de hardware de la máquina virtual debian2
Fuente: el autor (2011)

Otras configuraciones previas necesarias para asegurar el correcto funcionamiento del diseño propuesto es la configuración de los ficheros: /etc/hostname, /etc/hosts y /etc/network/interfaces.

Fichero /etc/hostname: se le asignó a cada nodo el nombre ast1 y ast2 dentro del fichero correspondiente a cada máquina.

Fichero /etc/network/interfaces: para cada nodo se configuró la interfaz de red correspondiente y para ast1, el cual será configurado como nodo primario, se configuró también la interfaz virtual correspondiente. La configuración de las interfaces de red para ast1 y ast2 se muestra en las figuras 26 y 27.



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
GNU nano 2.2.4 Fichero: /etc/network/interfaces

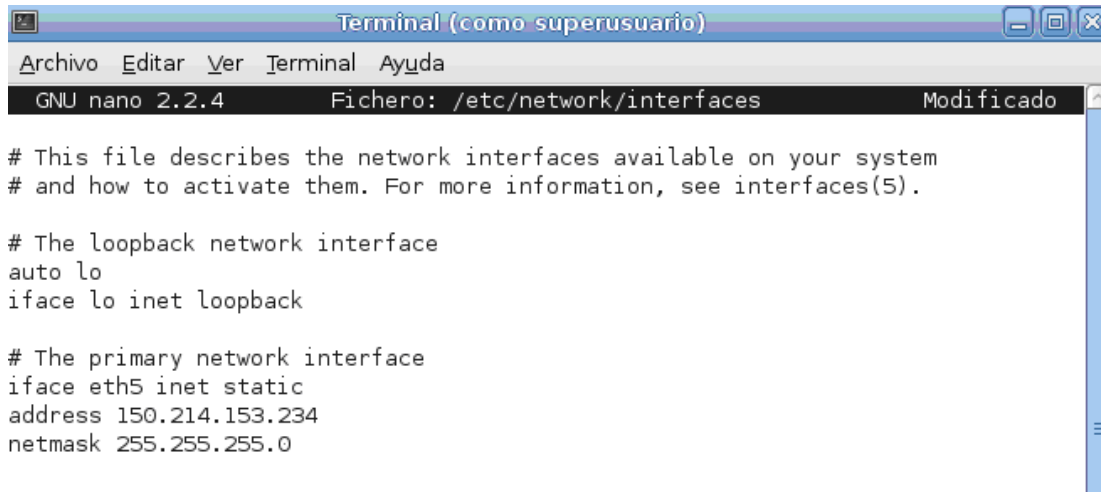
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth5
iface eth5 inet static
address 150.214.153.233
netmask 255.255.255.0

auto eth5:0
iface eth5:0 inet static
address 150.214.153.235
netmask 255.255.255.0
#NetworkManager#iface eth0 inet dhcp
```

Figura 26. Configuración del fichero /etc/network/interfaces para ast1
Fuente: el autor (2011)



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
GNU nano 2.2.4 Fichero: /etc/network/interfaces Modificado

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
iface eth5 inet static
address 150.214.153.234
netmask 255.255.255.0
```

Figura 27. Configuración del fichero /etc/network/interfaces para ast2
Fuente: el autor (2011)

Fichero/etc/hosts: luego de asignadas las direcciones IP a cada nodo se hicieron coincidir con los nombres asignados, agregando las siguientes líneas en este fichero para cada nodo:

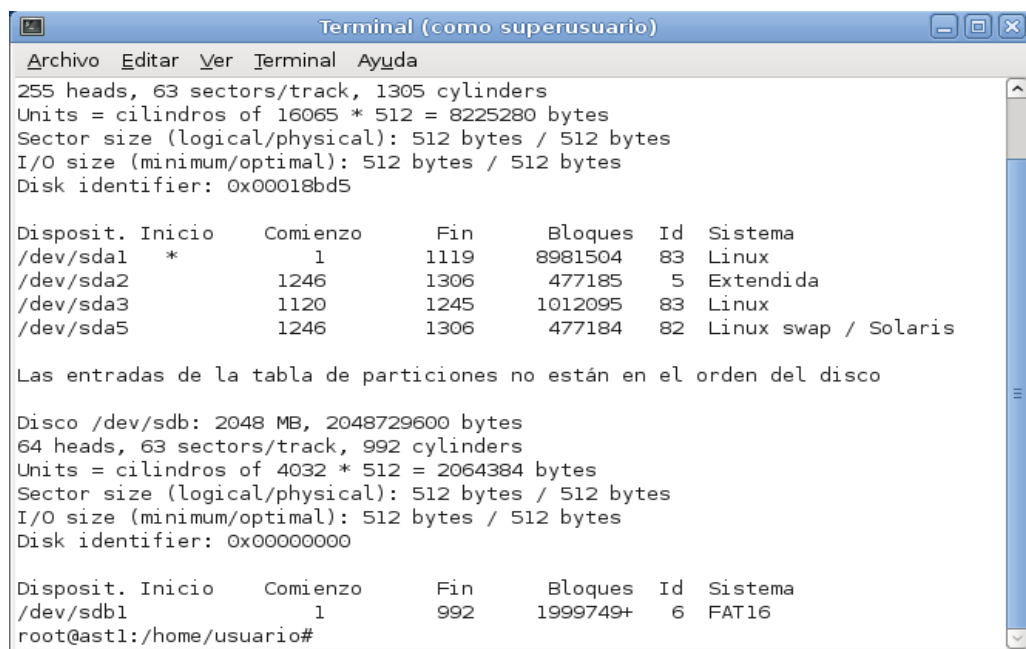
150.214.153.233 ast1

150.214.153.243 ast2

Luego de haber completado estos requisitos previos, se procedió con la instalación y configuración del soporte para Alta Disponibilidad de datos DRBD.

Configuración del soporte de Alta Disponibilidad en datos

La instalación y configuración de DRBD se realizó siguiendo la metodología expuesta en la descripción de la propuesta. Al momento de la instalación del sistema operativo se creó la partición extra en cada máquina, la cual se va a utilizar para la replicación de los datos. La estructura de las particiones del disco para ast1 se muestra en la figura 28 y la estructura de las particiones del disco de ast2 en la figura 29.



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
255 heads, 63 sectors/track, 1305 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00018bd5

Disposit. Inicio Comienzo Fin Bloques Id Sistema
/dev/sda1 * 1 1119 8981504 83 Linux
/dev/sda2 1246 1306 477185 5 Extendida
/dev/sda3 1120 1245 1012095 83 Linux
/dev/sda5 1246 1306 477184 82 Linux swap / Solaris

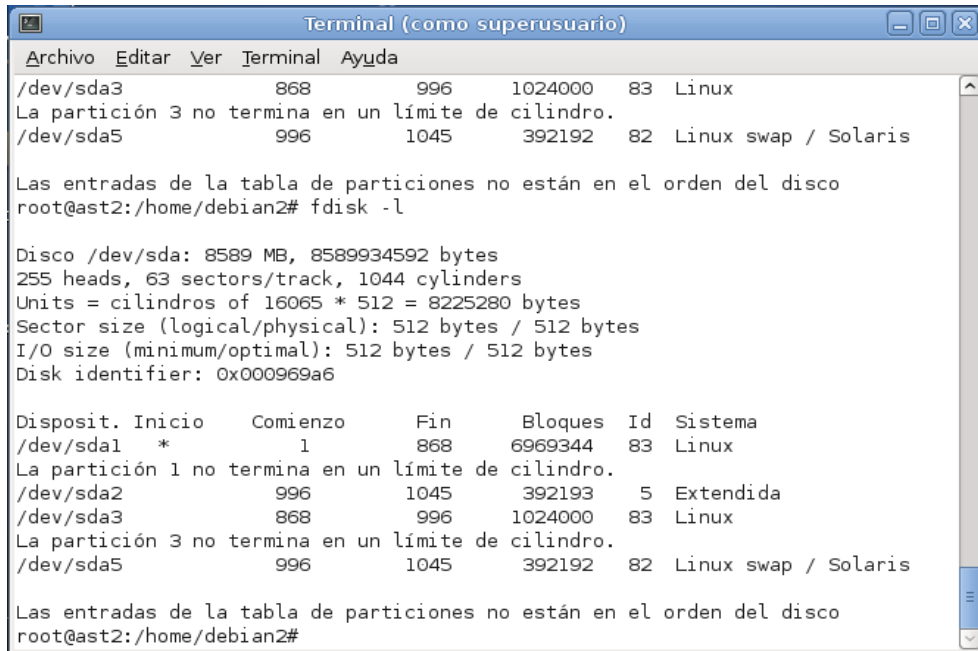
Las entradas de la tabla de particiones no están en el orden del disco

Disco /dev/sdb: 2048 MB, 2048729600 bytes
64 heads, 63 sectors/track, 992 cylinders
Units = cilindros of 4032 * 512 = 2064384 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disposit. Inicio Comienzo Fin Bloques Id Sistema
/dev/sdb1 1 992 1999749+ 6 FAT16
root@ast1:/home/usuario#
```

Figura 28. Estructura de particiones de disco de ast1

Fuente: el autor (2011)



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
/dev/sda3      868      996      1024000  83  Linux
La partición 3 no termina en un límite de cilindro.
/dev/sda5      996      1045      392192  82  Linux swap / Solaris

Las entradas de la tabla de particiones no están en el orden del disco
root@ast2:/home/debian2# fdisk -l

Disco /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000969a6

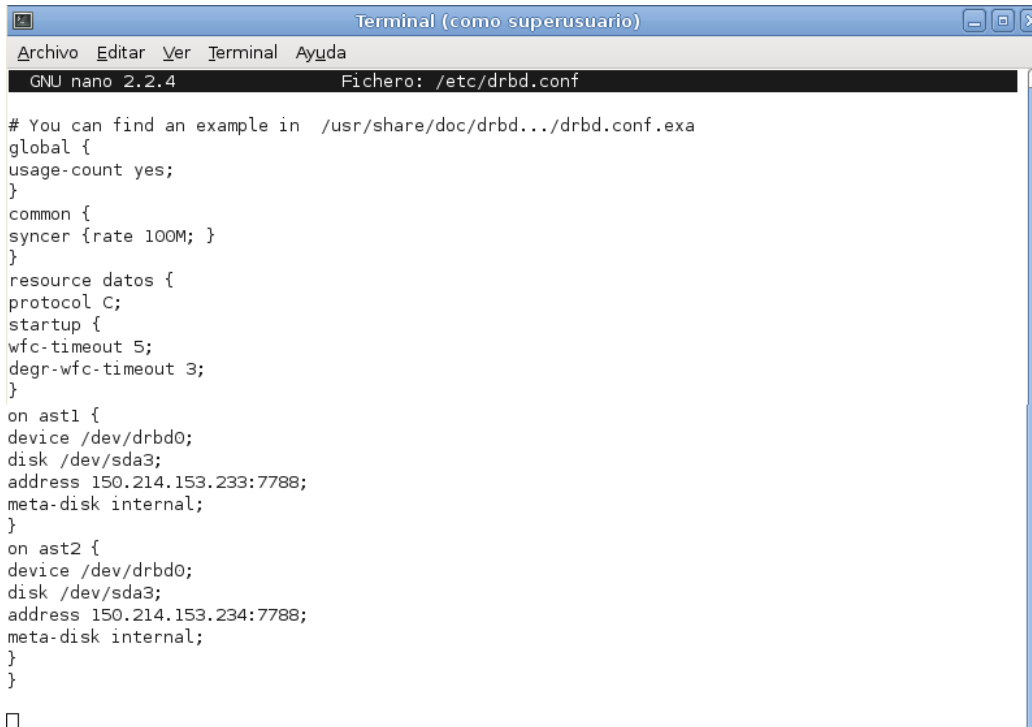
Disposit. Inicio    Comienzo      Fin          Bloques  Id Sistema
/dev/sda1 *          1              868         6969344  83 Linux
La partición 1 no termina en un límite de cilindro.
/dev/sda2            996           1045         392193   5 Extendida
/dev/sda3            868           996         1024000  83 Linux
La partición 3 no termina en un límite de cilindro.
/dev/sda5            996           1045         392192  82 Linux swap / Solaris

Las entradas de la tabla de particiones no están en el orden del disco
root@ast2:/home/debian2#
```

Figura 29. Estructura de particiones de disco de ast1

Fuente: el autor (2011)

Luego de descargar e instalar DRBD y las utilidades necesarias, se procedió a la configuración del fichero `/etc/drbd.conf`, el cual debe ser igual en ambos nodos. La configuración del fichero se muestra en la figura 30.



```
Terminal (como superusuario)
Archivo  Editar  Ver  Terminal  Ayuda
GNU nano 2.2.4      Fichero: /etc/drbd.conf

# You can find an example in /usr/share/doc/drbd.../drbd.conf.exa
global {
usage-count yes;
}
common {
syncer {rate 100M; }
}
resource datos {
protocol C;
startup {
wfc-timeout 5;
degr-wfc-timeout 3;
}
on ast1 {
device /dev/drbd0;
disk /dev/sda3;
address 150.214.153.233:7788;
meta-disk internal;
}
on ast2 {
device /dev/drbd0;
disk /dev/sda3;
address 150.214.153.234:7788;
meta-disk internal;
}
}

□
```

Figura 30. Configuración del fichero /etc/drbd.conf en ambos nodos
Fuente: el autor (2011)

En ambos nodos se creó el espacio para metadatos y la unidad virtual a la que se ha hecho referencia en el archivo de configuración *drbd.conf* , mediante las siguiente instrucciones:

```
#drbdadm create-md datos
#drbdadm attach datos
```

El paso siguiente fue iniciar el servicio DRBD en ambos nodos, para luego establecer la comunicación entre ellos y pudieran comenzar a sincronizarse:

```
#/etc/init.d/drbd start
#drbdadm connect datos
```

Finalmente se estableció el nodo ast1 como primario mediante la instrucción:

```
# cat /proc/drbd
```

Luego de instalado y configurado DRBD, se procedió a tomar los ficheros de configuración de Asterisk y los directorios donde se almacena la información de los usuarios para colocarlos en la partición virtual. Luego se crearon los enlaces desde estas nuevas ubicaciones hacia sus ubicaciones originales de manera que no se vea alterado el funcionamiento de Asterisk.

Luego de este procedimiento, la información los archivos de asterisk se mantendrá sincronizada periódicamente, de manera que de aquí en adelante cualquier nodo que presta el servicio contará con la información actualizada. Hasta aquí se garantiza la disponibilidad de los datos de Asterix, el siguiente paso es evaluar la alta disponibilidad des servicio Asterisk.

Configuración del soporte de Alta Disponibilidad en servicios

De igual manera que para la configuración del soporte de alta disponibilidad en datos, para la instalación y configuración del soporte en Alta Disponibilidad en Servicios se siguió la metodología explicada durante la descripción de la propuesta.

Luego de descargar e instala en ast1 y ast2r los componentes del proyecto LINUX-HA, se prosiguió con la configuración del fichero ha.cf en ambos nodos del cluster, teniendo en consideración la interfaz de red asignada a cada uno. Los valores modificados dentro del fichero en cada nodo fueron los siguientes:

debugfile /var/log/ha-debug

logfile /var/log/ha-log

logfacility local 0

keepalive 2

deadtime 20

warntime 10

initdead 20

udpport 694

bcast eth5 // para esta evaluación coincidieron las interfaces en ambos nodos

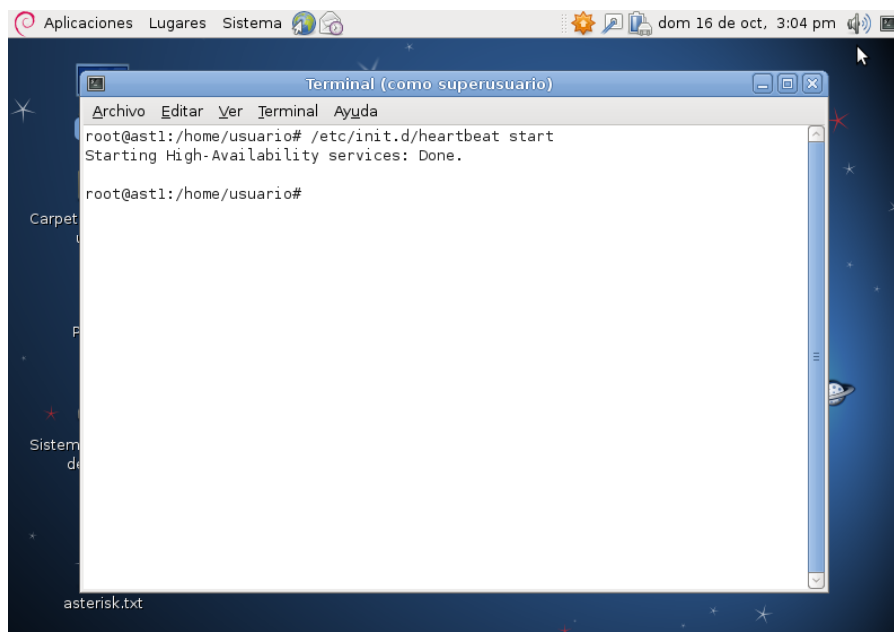
autofailback off

node ast1

node ast2

crm yes

A continuación se inició el servicio en ambos nodos, como se puede observar en las figuras 31 y 32.

A screenshot of a Linux desktop environment. The top panel shows the system menu with 'Aplicaciones', 'Lugares', and 'Sistema'. The system clock indicates 'dom 16 de oct, 3:04 pm'. A terminal window titled 'Terminal (como superusuario)' is open, displaying the following text:

```
Archivo Editar Ver Terminal Ayuda
root@ast1:/home/usuario# /etc/init.d/heartbeat start
Starting High-Availability services: Done.
root@ast1:/home/usuario#
```

The terminal window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Terminal', and 'Ayuda'. The desktop background is dark blue with a starry pattern and a small rocket icon in the bottom right corner. The text 'asterisk.txt' is visible in the bottom left corner of the terminal window.

Figura 31. Inicio del servicio de heartbeat en ast1

Fuente: El autor

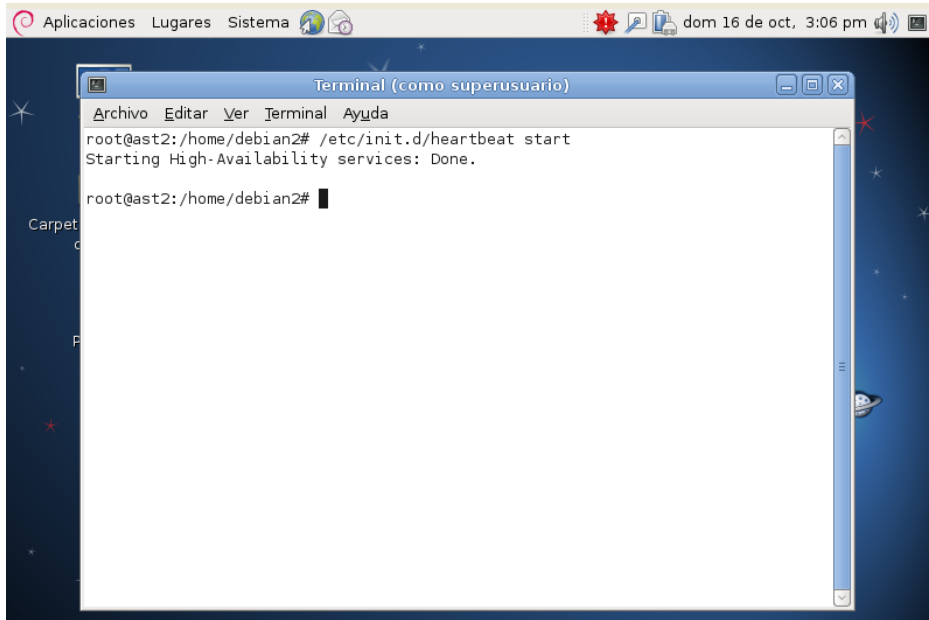
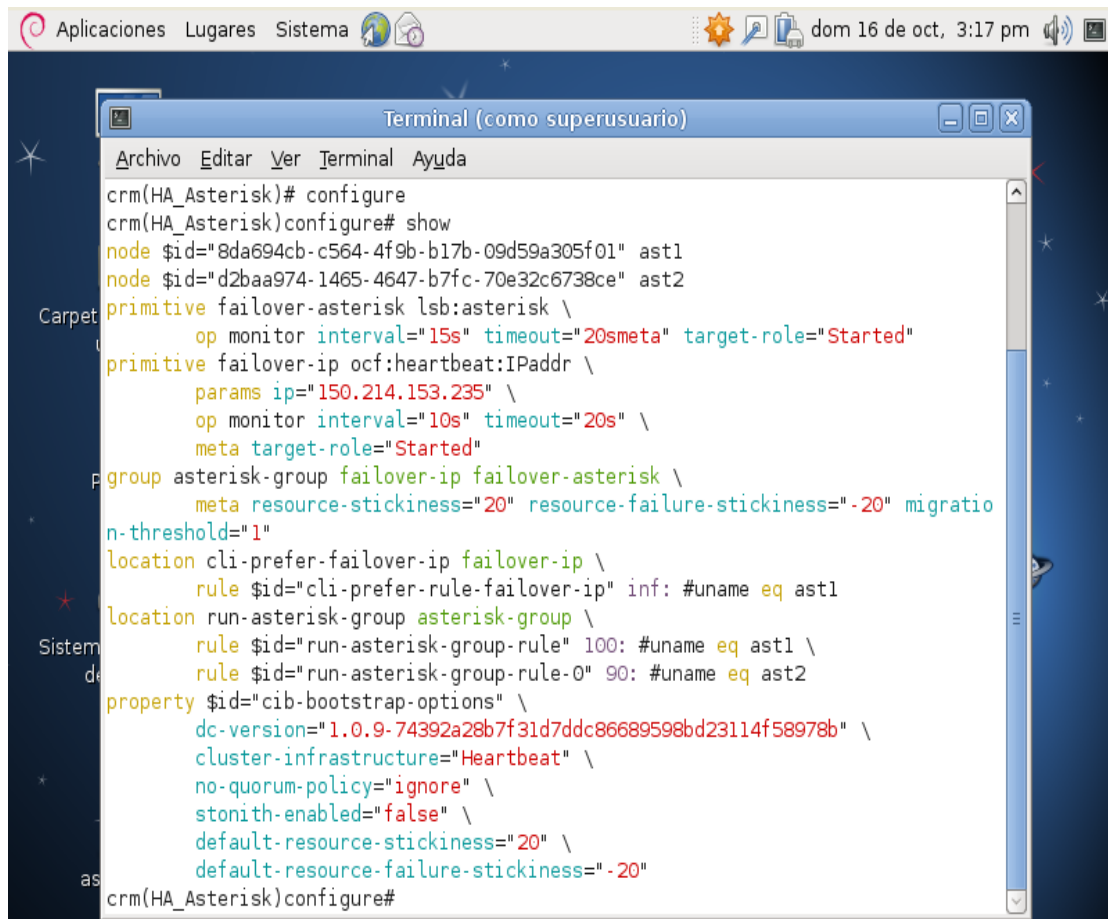


Figura 32. Inicio del servicio de heartbeat en ast2

Fuente: El autor

Por medio de crm, se configuró el fichero cib shadow en ast1. La figura 33 muestra la configuración del fichero cib. Allí se observa que fueron creados los recursos failover-ip y failover –asterisk y los cuales forman parte del grupo asterik-group.



```
crm(HA_Asterisk)# configure
crm(HA_Asterisk)configure# show
node $id="8da694cb-c564-4f9b-b17b-09d59a305f01" ast1
node $id="d2baa974-1465-4647-b7fc-70e32c6738ce" ast2
primitive failover-asterisk lsb:asterisk \
    op monitor interval="15s" timeout="20s" meta target-role="Started"
primitive failover-ip ocf:heartbeat:IPaddr \
    params ip="150.214.153.235" \
    op monitor interval="10s" timeout="20s" \
    meta target-role="Started"
group asterisk-group failover-ip failover-asterisk \
    meta resource-stickiness="20" resource-failure-stickiness="-20" migration-threshold="1"
location cli-prefer-failover-ip failover-ip \
    rule $id="cli-prefer-rule-failover-ip" inf: #uname eq ast1
location run-asterisk-group asterisk-group \
    rule $id="run-asterisk-group-rule" 100: #uname eq ast1 \
    rule $id="run-asterisk-group-rule-0" 90: #uname eq ast2
property $id="cib-bootstrap-options" \
    dc-version="1.0.9-74392a28b7f31d7ddc86689598bd23114f58978b" \
    cluster-infrastructure="Heartbeat" \
    no-quorum-policy="ignore" \
    stonith-enabled="false" \
    default-resource-stickiness="20" \
    default-resource-failure-stickiness="-20"
crm(HA_Asterisk)configure#
```

Figura 33. Configuración del fichero cib shadow

Fuente: El autor

Finalmente, con el comando commit desde crm, se difundieron los cambios hacia ast2, y el cluster de Alta disponibilidad en servicios quedó configurado y activo en este punto.

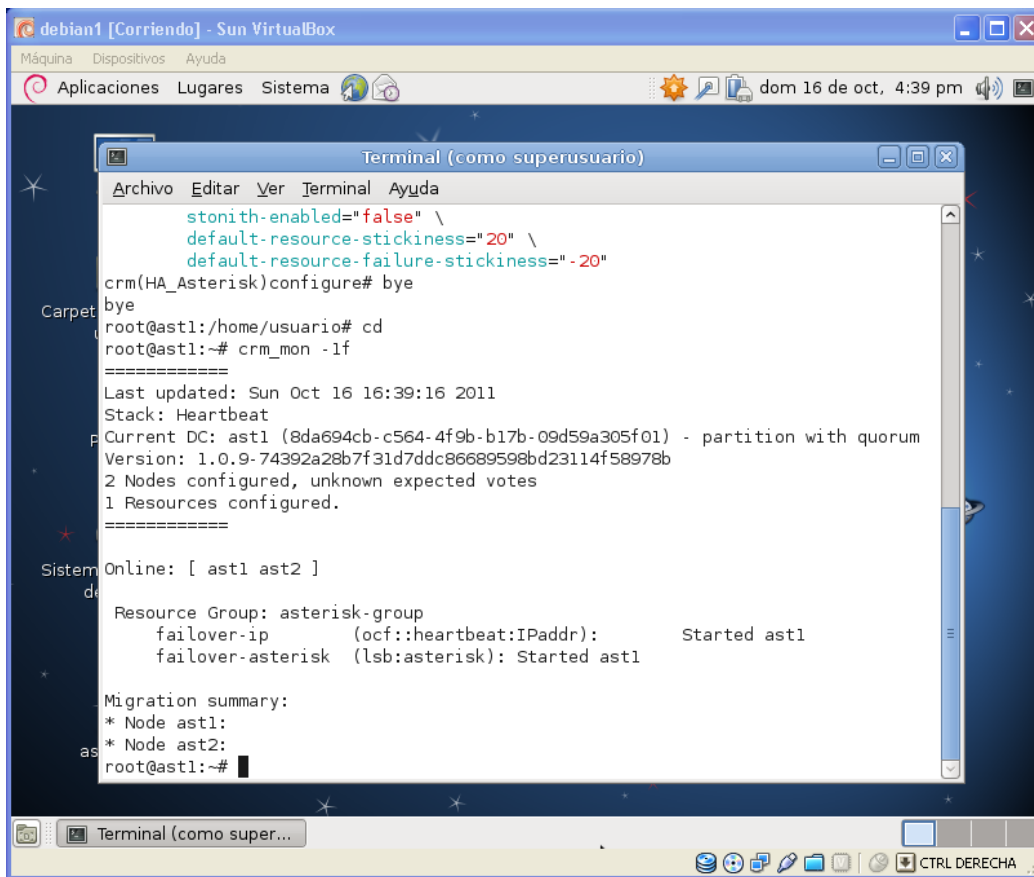
Prueba del Cluster Virtual de Alta Disponibilidad en Servicios

Para probar la configuración del Cluster de Alta Disponibilidad diseñado como parte de la propuesta de este trabajo, se analizó el comportamiento del mismo ante tres sucesos diferentes que provocarían la migración del grupo de recursos de un nodo

a otro: *apagado del nodo activo, caída del servicio Asterisk y caída del servicio Heartbeat*.

Apagado del nodo activo

Para esta prueba, simplemente se procedió a apagar la maquina virtual en la cual estaban corriendo los recursos (ast1) y observar si se cumple la migración de los recursos al otro nodo. Pero antes se verificó el funcionamiento del cluster y que los recursos realmente estuvieran corriendo en ast1. Mediante el comando `crm_mon -1f` se puede observar el estado actual del cluster como se muestra en la figura 34.



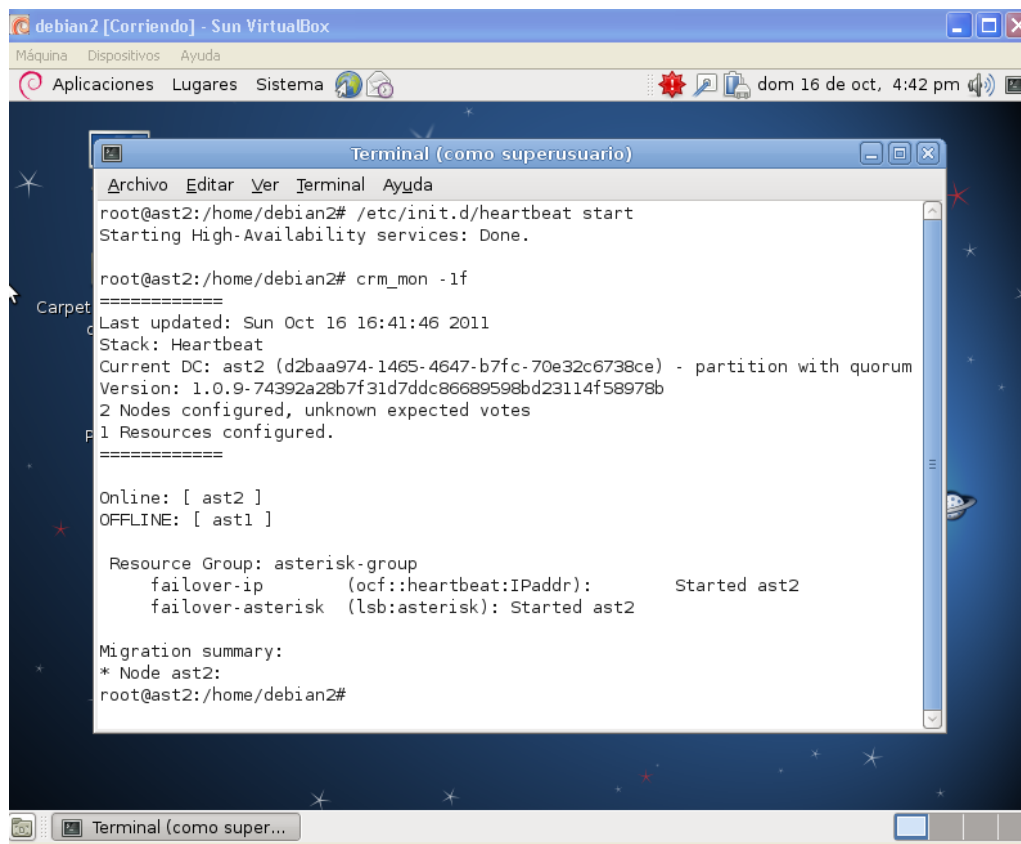
```
stonith-enabled="false" \  
default-resource-stickiness="20" \  
default-resource-failure-stickiness="-20"  
crm(HA_Asterisk)configure# bye  
bye  
root@ast1:/home/usuario# cd  
root@ast1:~# crm_mon -1f  
=====  
Last updated: Sun Oct 16 16:39:16 2011  
Stack: Heartbeat  
Current DC: ast1 (8da694cb-c564-4f9b-b17b-09d59a305f01) - partition with quorum  
Version: 1.0.9-74392a28b7f31d7ddc86689598bd23114f58978b  
2 Nodes configured, unknown expected votes  
1 Resources configured.  
=====  
Online: [ ast1 ast2 ]  
  
Resource Group: asterisk-group  
failover-ip (ocf::heartbeat:IPaddr): Started ast1  
failover-asterisk (lsb:asterisk): Started ast1  
  
Migration summary:  
* Node ast1:  
* Node ast2:  
root@ast1:~#
```

Figura 34. estatus del cluster antes del apagado de ast1

Fuente: El autor

Se procedió al apagado del nodo ast1, y luego se utilizó nuevamente el comando `crm_mon -lf`, ahora en el nodo ast2.

Al ejecutar este comando se pudo observar la migración exitosa de los recursos a ast2 luego del apagado del nodo ast1, como se observa en la figura 35, el nodo ast1 está fuera de línea y los recursos se encuentran activos ahora en el nodo ast2



```
root@ast2:/home/debian2# /etc/init.d/heartbeat start
Starting High-Availability services: Done.

root@ast2:/home/debian2# crm_mon -lf
=====
Last updated: Sun Oct 16 16:41:46 2011
Stack: Heartbeat
Current DC: ast2 (d2baa974-1465-4647-b7fc-70e32c6738ce) - partition with quorum
Version: 1.0.9-74392a28b7f31d7ddc86689598bd23114f58978b
2 Nodes configured, unknown expected votes
1 Resources configured.
=====

Online: [ ast2 ]
OFFLINE: [ ast1 ]

Resource Group: asterisk-group
    failover-ip      (ocf::heartbeat:IPaddr):      Started ast2
    failover-asterisk (lsb:asterisk): Started ast2

Migration summary:
* Node ast2:
root@ast2:/home/debian2#
```

Figura 35. Estatus del cluster luego del apagado de ast1

Fuente: El autor

También, puede observarse en la configuración de las interfaces de red, como es levantada una interfaz con la dirección IP configurada como virtual mediante la salida del comando `ifconfig`, como se muestra en la figura 36.

```

TX packets:162 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:55786 (54.4 KiB) TX bytes:17904 (17.4 KiB)
Interrupt:10 Base address:0xd020

eth5  Link encap:Ethernet  HWaddr 08:00:27:f0:e8:b0
      inet addr:150.214.153.234  Bcast:150.214.153.255  Mask:255.255.255.0
      inet6 addr: fe80::a00:27ff:fe0:e8b0/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:735152 errors:2651 dropped:0 overruns:0 frame:0
      TX packets:88189 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:1087380894 (1.0 GiB) TX bytes:9244428 (8.8 MiB)
      Interrupt:9 Base address:0xd240

eth5:1 Link encap:Ethernet  HWaddr 08:00:27:f0:e8:b0
      inet addr:150.214.153.235  Bcast:150.214.153.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      Interrupt:9 Base address:0xd240

```

Figura 36. Configuración de red de ast2 luego del apagado de ast1

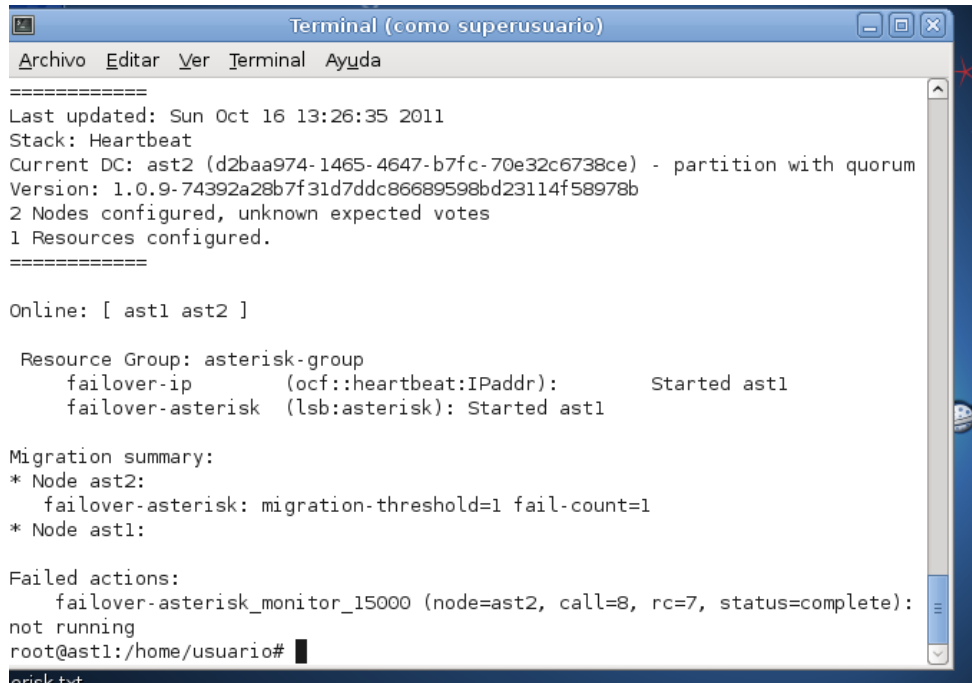
Fuente: El autor

Caída del servicio Asterisk

Esta prueba consiste en provocar la caída de Asterisk en el nodo activo y observar como reacciona Heartbeat ante este evento. Para completar esta segunda prueba, se encendió nuevamente el nodo ast1, pero los recursos continúan activos en el nodo ast2, ya que como se señaló en la descripción de la propuesta, la variable `auto_failback` fue configurado en `off`, para evitar migraciones consecuentes y no administradas. Para simular la caída del servicio, se detuvo manualmente haciendo uso del script Asterisk ubicado en el directorio `/etc/init.d`, por medio de la siguiente instrucción desde la consola de root de ast2:

```
root@ast2:# /etc/init.d/Asterisk stop
```

La caída de Asterisk es detectada por Heartbeat al consumirse el timeout asociado a la operación `monitor` del recurso `failover-asterisk`, aumentando en 1 y como este es el límite de fallos establecidos en este momento en heartbeat, los servicios son migrados al otro nodo del cluster como se observa en la figura 37



```
Terminal (como superusuario)
Archivo  Editar  Ver  Terminal  Ayuda
=====
Last updated: Sun Oct 16 13:26:35 2011
Stack: Heartbeat
Current DC: ast2 (d2baa974-1465-4647-b7fc-70e32c6738ce) - partition with quorum
Version: 1.0.9-74392a28b7f31d7ddc86689598bd23114f58978b
2 Nodes configured, unknown expected votes
1 Resources configured.
=====

Online: [ ast1 ast2 ]

Resource Group: asterisk-group
  failover-ip      (ocf::heartbeat:IPAddr):      Started ast1
  failover-asterisk (lsb:asterisk): Started ast1

Migration summary:
* Node ast2:
  failover-asterisk: migration-threshold=1 fail-count=1
* Node ast1:

Failed actions:
  failover-asterisk_monitor_15000 (node=ast2, call=8, rc=7, status=complete):
not running
root@ast1:/home/usuario#
```

Figura 37. Estatus del cluster luego de la parada del servicio Asterisk en ast2

Fuente: El autor

Caída del servicio Heartbeat

La última prueba a la que se sometió el cluster de Alta Disponibilidad de Servicios configurado consistió en simular la caída del servicio Heartbeat en el nodo activo (ast2 para el momento de esta prueba). Para ello se procedió de manera similar a la prueba anterior, deteniendo del servicio Heartbeat manualmente en ast2:

```
root@ast2:# /etc/init.d/heartbeat stop
```

Luego de detener el servicio Heartbeat en el nodo activo, los recursos son migrados al otro nodo y el nodo en el cual se detuvo el servicio es colocado como fuera de línea. El estatus del cluster luego de la parada de Heartbeat en el nodo ast2 se muestra en la figura 38.

```
Migration summary:
* Node ast1:
root@ast1:/home/usuario# crm_mon -lf
=====
Last updated: Sun Oct 16 17:13:16 2011
Stack: Heartbeat
Current DC: ast1 (8da694cb-c564-4f9b-b17b-09d59a305f01) - partition with quorum
Version: 1.0.9-74392a28b7f31d7ddc86689598bd23114f58978b
2 Nodes configured, unknown expected votes
1 Resources configured.
=====

Online: [ ast1 ]
OFFLINE: [ ast2 ]

Resource Group: asterisk-group
    failover-ip      (ocf::heartbeat:IPAddr):      Started ast1
    failover-asterisk (lsb:asterisk): Started ast1

Migration summary:
* Node ast1:
root@ast1:/home/usuario#
```

Figura 38. Estatus del cluster luego de la parada del servicio heartbeat en ast2

Fuente: El autor

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Por medio del desarrollo de esta investigación, quedó demostrado que existe la necesidad de incorporar Alta Disponibilidad al actual Sistema de Telefonía IP de la sede administrativa principal del IPASME. Esto se logró mediante el estudio de las dimensiones de la variable de estudio (Alta Disponibilidad del Sistema De Telefonía IP) en función de los indicadores definidos por cada dimensión. La combinación del juicio de los usuarios ordinarios del sistema y los usuarios expertos permitió que el diagnóstico fuera mucho más completo.

Posteriormente, se determinó que incorporar Alta Disponibilidad al Sistema de Telefonía IP de la Sede Administrativa Principal del IPASME es viable desde el punto de vista técnico, operativo, institucional y económico.

Es posible además, plantear diversos esquemas de Alta Disponibilidad que se integren al actual Sistema de Telefonía IP de la Sede Administrativa Principal del IPASME.

Partiendo de los diversos esquemas planteados, fue posible proponer un diseño de Alta Disponibilidad que se integre al actual Sistema de Telefonía IP objeto del estudio en función de la optimización del hardware y el uso de tecnologías libres.

De acuerdo a la evaluación hecha del diseño propuesto, se pudo comprobar que garantiza la Alta Disponibilidad del servicio de Telefonía IP de los casos más comunes de incidencia.

Recomendaciones

Una vez concluida la presente investigación se recomienda;

1. Implantar la propuesta de Alta Disponibilidad diseñada en este trabajo, para garantizar la continuidad del servicio de Telefonía IP en la sede administrativa principal del IPASME.
2. Para la implantación del diseño propuesto, considerar las múltiples soluciones para la virtualización de servidores, de manera que se puedan aprovechar mejor los recursos de hardware y puedan reducirse los costos de implantación del diseño propuesto.
3. Es importante estudiar y definir todos aquellos elementos a nivel de hardware que pudieran estar redundados dentro del sistema de Telefonía IP (tolerancia a fallos), tales como: fuentes de poder de los servidores y tarjetas de interfaz de red, entre otros componentes físicos que también llegarían a comprometer la disponibilidad del servicio en caso de fallos
4. Al implantar el diseño propuesto, someterlo a pruebas de saturación de llamadas que permitan evaluar el rendimiento del Sistema.
5. De acuerdo a los resultados obtenidos durante las pruebas de saturación del Sistema, se puede pensar en desarrollar un diseño de Alto Rendimiento que pueda incorporarse al diseño propuesto en este trabajo
6. Considerar la migración del Sistema de Telefonía IP de la sede administrativa principal del IPASME hacia una única plataforma basada en estándares abiertos y que incorpore soporte de Alta Disponibilidad y Alto rendimiento, atendiendo no sólo a estas recomendaciones, sino también a las recomendaciones hechas por García, K. (2009).

REFERENCIAS BIBLIOGRÁFICAS

Ary, W. 1996. *Metodología de la Investigación*. Ediciones Roalg. Madrid. España.

Asamblea Nacional de la República Bolivariana de Venezuela. 2000. leyes. Formato digital. URL:

http://www.asambleanacional.gov.ve/index.php?option=com_leyesficha&Itemid=239&lang=es (Consulta: Noviembre, 2010)

Balestrini, M. 1998. *Cómo se elabora el Proyecto de Investigación en Venezuela*. Ediciones Consultores Asociados, Servicio Editorial. Caracas. Venezuela.

Balestrini, M. 2002. *Cómo se elabora el Proyecto de Investigación*. Ediciones Consultores Asociados, Servicio Editorial. Caracas. Venezuela.

Barberán, J. 2009. *Implantación de un Sistema de VoIP basado en Asterisk*. Tesis de grado. Universidad Peruana de Ciencias Aplicadas. Lima.

Casas, S. y Reniaga, H. 2009. *Esquema de redundancia y distribución de carga de alta disponibilidad para la prestación de Telefonía IP usando SIP*. Cuarto Congreso Colombiano de Computación. Sociedad Colombiana de Computación. Cali.

Colobran, M. y Otros. 2008. *Administración de Sistemas Operativos en red*. Editorial UOC. Barcelona.

Figuroa, W. 2008. *Implantación de una Central Telefónica Privada de VoIP bajo Trixbox 2.2 de Alta Disponibilidad para la Gerencia de Direccionamiento Tecnológico de CANTV*. Trabajo de Grado. Universidad Católica Andrés Bello. Caracas.

García, K. 2009. *Diseño e Implantación de una Solución de Voz sobre IP basada de Software Libre para las unidades del PASME*. Tesis de grado. UNEXPO. Caracas.

Guadamuz, E. 2007. *Diseño de Sistemas de VoIP con Asterisk*. Tesis de Grado. Universidad de Costa Rica. Costa Rica.

Hernández, R. y Otros. 2007. *Metodología de la Investigación*. McGraw-Hill, Interamericana de Editores. México.

- Lillo, S. 2007. Virtualización: mayor eficiencia y flexibilidad. Desayunos Tecnológicos. Coriolis IT Solutions. Chile.
- Martín, V. y Fuentes, A. 2009. Estudio e Implantación de un Sistema de Alta Disponibilidad en RedIRIS. RedIRIS. Boletín N° 85-86.
- Méndez, F. 2009. Diseño en Implementación de un Sistema de VoIP de Alta Disponibilidad y Alto Rendimiento. Tesis de Grado. Universidad de Almería. Almería.
- Rodríguez, R. 2006. Metodología para el Diseño de una red VoIP de Alto Tráfico y QoS. Tesis de Grado Maestría. Instituto Politécnico Nacional. Tijuana.
- Rodríguez, T, y Díaz, J. 2007. Aplicaciones sobre una red de Telefonía IP. Trabajo de grado. Universidad Técnica Federico Santa María. Chile.
- Sierra, A. 2008. Instalación de un Sistema de VoIP Corporativo basado en Asterisk. Tesis de Grado. Universidad Politécnica de Cartagena. Cartagena.
- Tamayo, M. 1999. El proceso de la investigación científica. Editorial Limusa S.A. México.
- Comunidad del Proyecto DRBD. The DRBD USER'S GUIDE (2009). URL: <http://www.drbd.org/user-guide/> (Consulta: Agosto, 2011)
- Universidad Centroccidental "Lisandro Alvarado". 2002. Manual para la Elaboración del Trabajo Conducente a Grado Académico de Especialización, Maestría y Doctorado. Sesión Ordinaria N° 1353.
- Villar, E. 2010. Virtualización de servidores de Telefonía IP en GNU/LINUX. Tesis de Grado. Universidad de Almería. Almería
- 3CX. 2009. VoIP-SIP. Home page. URL: <http://www.3cx.es/voip-sip> (Consulta: Agosto, 2010).

ANEXOS

ANEXO A.
CURRICULUM VITAE DEL AUTOR

ANEXO B.
CUESTIONARIO 01

ANEXO C.
CUESTIONARIO 02

ANEXO D.
TABLA DE RECOPIACIÓN DE DATOS

ANEXO E.
FORMATOS DE VALIDACION DE LOS INSTRUMENTOS

.

ANEXO F.
CALCULO DE LA CONFIABILIDAD DE LOS INSTRUMENTOS
MEDIANTE EL ALPHA DE CRONBACH

