UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

ALTA DISPONIBILIDAD EN SERVIDORES WEB BAJO LINUX CASO: SERVIDOR WEB DE LA FACULTAD DE INGENIERÍA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :

JOSÉ DE JESÚS RAMÍREZ PICHARDO

DIRECTORA DE TESIS: M. C. MARÍA JAQUELINA LÓPEZ BARRIENTOS

CD. UNIVERSITARIA, MÉXICO D. F. OCTUBRE 2004

Contenido

AGRADECIMIENTOS

*A mi **Padre**, a quien día a día encomiendo mi espíritu en sus manos.*

A mis Padres, María y José Luis, por su amor y su apoyo continuo e incondicional.

A mis Hermanos, Josser y Luis, por su grandiosa ayuda.

A mis pequeñas Rubit, Perla y Cristal, por ser mi motivación.

A toda mi Familia, porque de alguna manera siempre están ahí en el justo momento.

A la Universidad Nacional Autónoma de México, fuente infinita de saber, por impulsar mi formación universal.

A toda la Comunidad de la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería, por la oportunidad de crecer como profesionista y persona.

A la Maestra Jaquelina, por su valioso apoyo y su asertiva guía y enseñanzas.

A mis compañeros de Generación, por estos años de lucha y entrega que compartimos.

A mis grandiosos amigos, por su apoyo sincero y desinteresado y sus valiosos consejos que me guían día con día.

A todas aquellas grandes personas con las cuales, de alguna forma, he escrito una página en el libro de mi vida.

GRACIAS

INTRODUCCIÓN

INTRODUCCIÓN

i.1 PLANTEAMIENTO GENERAL

Hace no muchos años, la mayoría de los sitios que formaban parte de la red de redes, Internet, estaban compuestos por páginas estáticas, es decir, el contenido presentado a un usuario, era el mismo para todos aquellos que la visitaran. En la actualidad, los sitios son dinámicos, puesto que, para muchas aplicaciones en Internet, es necesario que el contenido de sus páginas cambie de acuerdo al usuario que las visita. Por ejemplo, un sitio de comercio electrónico generalmente mantiene información sobre los clientes, y posteriormente ésta misma información es usada para mostrar un contenido personalizado.

El dinamismo de un sitio en Internet requiere que el servidor donde está alojado lleve a cabo un trabajo adicional, por ejemplo: la conexión a un servidor de base de datos local o remota para proveer a las aplicaciones de los datos necesarios para su correcto funcionamiento.

El uso de Tecnologías de la Información incrementa la presión de la competencia, la globalización de los servicios, el comercio electrónico, así como la demanda de accesos a la información de manera rápida, fácil y continua.

Conclusión: las empresas, los gobiernos, las instituciones de educación y de salud y la sociedad en general se mueven hacía entornos más dinámicos y continuamente en línea.

Tanto en el mundo empresarial, como en el mundo académico, existen ciertas aplicaciones que dada su naturaleza deben proporcionar un servicio sin interrupciones las 24 horas del día, los 7 días de semana y los 365 días del año, ya sea para dar soporte interno (procesos de contabilidad, control de personal, desarrollo, etc.) como para ofrecer servicios a través de Internet (comercio electrónico, correo, portales, etc.).

En 1995, compañías como *Oracle Corp.* y *Datamation*, en estudios independientes, revelaron que una mediana empresa tiene un pérdida entre 80,000 y 350,000 dólares por hora de interrupción no planeada de sus servicios informáticos.

Tras el atentado en 1993 contra el *World Trade Center* (*Oklahoma, USA*), 145 de las 350 empresas que allí hospedaban toda su infraestructura informática, tuvieron que cerrar sus puertas.

Las interrupciones no planeadas son simplemente inaceptables para una empresa que quiera mantener los compromisos con sus clientes.

Algunas de las posibles consecuencias de una interrupción no planeada son las siguientes (Ver Tabla i.1):

Pérdida de clientes
Pérdida de oportunidades
Pérdida de capacidad
Trabajo perdido o improductivo
Costos de restauración
Penalizaciones
Mala publicidad

Tabla i.1 Consecuencias de la falta de disponibilidad

i.2 CONTEXTO DEL PROBLEMA

Lo expuesto anteriormente, es el panorama general de todo sistema conectado a Internet, sea un servidor de comercio electrónico, entretenimiento, gobierno, salud o educación.

La Facultad de Ingeniería, en respuesta a la creciente globalización, que en cuanto a Tecnologías de la Información se refiere, desde hace unos años cuenta con su propio sitio: *www.ingenieria.unam.mx*

El Servidor Web en estudio proporciona los siguientes servicios: espacio físico en disco duro, accesos a bases de datos, y la posibilidad de ejecutar programas dinámicos escritos en PHP. Su importancia radica en que aloja, entre otros sitios, la página de la Facultad de Ingeniería. Organismos internos y personalidades que forman parte de la comunidad de la Facultad de Ingeniería también cuentan con un sitio Web.

¿Por qué motivo es importante que la página de la Facultad de Ingeniería este siempre disponible? El contenido del sitio incluye información sobre los funcionarios, la organización, las divisiones, carreras, Posgrado, Educación Continua, personal académico, asuntos estudiantiles, becas y apoyos académicos, servicios, etc. Cuenta también con una Biblioteca Digital con documentos técnicos de temas especiales de diversas carreras de la Facultad. Otros organismos internos de esta Facultad tienen alojado su sitio Web, tales como el Comité Asesor de Cómputo, la Unidad de Servicios de Cómputo Académico, la Coordinación de Proyectos Académicos y de Evaluación Educativa, la Tuna de la Facultad de Ingeniería, Sociedad de Alumnos de Ingeniería Minas y Metalurgistas, Servicio de Correo Electrónico para Académicos de la Facultad de Ingeniería, Coro de la Facultad de Ingeniería, sólo por mencionar algunos.

La información expuesta así como los sistemas que se ejecutan en el servidor son de vital importancia para la comunidad, y su disponibilidad debe ser un asunto a tomar seriamente.

A la necesidad de proveer un servicio sin interrupciones no planeadas se le conoce como **Alta Disponibilidad, *High Availability* ó HA**.

La principal técnica para asegurar este servicio de seguridad es la redundancia, que consiste en replicar las zonas críticas del sistema, teniendo una unidad activa y al menos una copia inactiva que, tras el fallo de la principal, sea capaz de retomar su labor en un punto muy cercano en el que aquella falló, en el menor tiempo posible y de forma lo más transparente para el usuario.

Entre las posibles causas de interrupciones no planeadas o falta de disponibilidad tenemos las siguientes:

Fallos de red (comunicaciones).
Fallos en el *hardware.*
Fallos en el sistema operativo.
Fallos en aplicaciones (Errores y rendimiento de las mismas).
Errores humanos.
Desastres naturales.

Tabla i.2 Causas de las Interrupciones no Planeadas

i.3 OBJETIVO

El objetivo que se quiere alcanzar al realizar este trabajo es: **IMPLEMENTAR UNA HERRAMIENTA DE SEGURIDAD CUYA MISIÓN SEA EVITAR PERDER EL SERVICIO ANTE CUALQUIER FALLA CAUSADA POR EL EQUIPO, SISTEMA OPERATIVO, APLICACIONES, FALTA DE ENERGÍA, DESASTRE NATURAL Y ANTE CUALQUIER FACTOR QUE ATENTE CONTRA LA DISPONIBILIDAD DEL SERVIDOR WEB.**

Las metas intermedias para alcanzar el objetivo son las siguientes:

Analizar todos los aspectos involucrados en la falta de disponibilidad de los servicios proporcionados por el servidor.

I*dentificar* las amenazas que atenten contra la disponibilidad de los servicios.

Diseñar una herramienta que contrarreste los efectos de dichas amenazas y proporcione Alta Disponibilidad en los servicios con base en técnicas de *clustering* en Linux.

Desarrollar la herramienta que brinde Alta Disponibilidad.

Implementar esta herramienta como mecanismo de seguridad y de Alta Disponibilidad.

i.4 JUSTIFICACIÓN

La disponibilidad, en el contexto de Tecnologías de la Información, es una medida relativa a la preparación para la utilización de un sistema informático, mientras que la fiabilidad es una medida relativa a la capacidad para mantenerse operativo en el tiempo. Ambas propiedades están englobadas dentro de una propiedad mucho más amplia, la confiabilidad, que también incluye aspectos de seguridad, confidencialidad e integridad de datos.

Todas ellas conllevan un elevado aumento del costo invertido en *hardware* y *software*. Al igual que ocurre en el mundo del deporte, realizar una carrera de 100 metros en menos de 11 segundos es relativamente fácil para un deportista de elite. A partir de aquí, cada décima de segundo de menos será fruto de duras horas de entrenamiento. En el caso que nos ocupa, eligiendo correctamente los componentes de nuestro sistema es relativamente fácil alcanzar un sistema disponible el 98% del tiempo. El paso del 98 al 99 por ciento, y de aquí al 99.999% (lo que los analistas-diseñadores de Sistemas de Alta Disponibilidad y Tolerantes a Fallos estadounidenses llaman los cinco nueves del tiempo de funcionamiento) es una tarea compleja que lleva aparejado un aumento exponencial del costo total del sistema.

Debido a la naturaleza del servicio (aplicaciones críticas), actualmente existen dos soluciones:

Sistemas Tolerantes a Fallos (Fault tolerant o FT): En los cuales, el servicio esta siempre activo. Todos los recursos necesarios están replicados para ofrecer una disponibilidad continua en caso de fallo.

Sistemas de Alta Disponibilidad: También está basada en la replicación de elementos, se replican aquellos recursos imprescindibles y se gestiona, mediante *software*, el fallo de los recursos y su recuperación. Naturalmente, si hablamos de replicar servidores, hablaremos de un *Cluster* de Alta Disponibilidad.

El problema de los Sistemas Tolerantes a Fallos es que son extremadamente caros y normalmente no hay presupuesto. Además suelen ser soluciones cerradas, totalmente dependientes de la empresa contratada. Se suele poner un Servidor Tolerante a Fallos, varias interfaces de red, con tomas de alimentación redundantes y climatización especial.

Los Sistemas de Alta Disponibilidad, intentan obtener prestaciones cercanas a los Sistemas Tolerantes a Fallos, pero a un precio mucho más interesante. Esta es una opción, que la ha hecho crecer en importancia dentro del mundo empresarial.

Evaluando estas condiciones, el presente trabajo contempla el empleo de un Sistema de Alta Disponibilidad.

Actualmente GNU/Linux es considerado como un sistema operativo estable (no infalible), y es una excelente elección para ser instalado en servidores que prestan servicios de Internet.

En general, en el mercado de GNU/Linux, los servicios de Alta Disponibilidad son la clave a la hora de vender soluciones para servidores de misión crítica. Prácticamente todos los proveedores de GNU/Linux poseen su propio *software* de Alta Disponibilidad que posibilita a los clientes tener un Sistema Tolerante a Fallos a un precio razonable.

Como alternativa, los administradores podrían organizar planes de acción a tomar ante las posibles caídas del sistema, indicando qué debe hacerse, cómo y por quién. Pero si la persona con los conocimientos necesarios no se encuentra disponible para llevar a cabo las acciones o bien comete un error, nuestro negocio estará en peligro. Las soluciones de Alta Disponibilidad reemplazan este esquema manual propenso a errores, mediante una serie de herramientas automatizadas que realizan las mismas tareas de recuperación en forma confiable y en menor tiempo.

Gracias a la disponibilidad del sistema operativo GNU/Linux y a los beneficios de la licencia GPL, es posible contar con un entorno para investigar, desarrollar y liberar aplicaciones. De esta forma, una empresa tiene la posibilidad de proveer soluciones personalizadas a sus clientes y brindar un mejor soporte, con la posibilidad de desempeñar un rol más activo que el de simple distribuidor de aplicaciones empaquetadas.

En el mercado existen muchas herramientas y Sistemas de Alta Disponibilidad, listas para implementarse. Sin embargo, éstas soluciones no se adaptan completamente a las necesidades ni satisfacen plenamente los requerimientos del cliente. El presente trabajo tiene la finalidad de analizar, diseñar, desarrollar e implementar un Sistema de Alta Disponibilidad que cubra plenamente los requerimientos tanto de funcionalidad como de seguridad, empleando los recursos necesarios y suficientes para dicho fin.

i.5 ORGANIZACIÓN DEL DOCUMENTO

Fundamentalmente, la organización del documento esta basada en la metodología a emplear que es la siguiente:

Antecedentes y marco teórico: Consistirá en analizar la base teórica que envuelve a los Sistemas de Alta Disponibilidad, elementos que los componen, dinámica o funcionamiento, las problemáticas que resuelve, así como un análisis de las soluciones existentes para garantizar la Alta Disponibilidad de un sistema informático.

Esquema de Seguridad del Servidor Web: Se realizará un análisis de la seguridad del entorno y del propio Servidor Web, con el fin de exponer aspectos que podrían atentar la integridad, confidencialidad y sobre todo, la disponibilidad del servicio.

Análisis de la información arrojada por el Esquema de Seguridad en relación a la Disponibilidad: Consistirá en observar la problemática y a partir de los datos arrojados por mecanismos como el Esquema de Seguridad, se determinarán todos los elementos involucrados y se investigarán los mecanismos y las técnicas que contrarresten sus consecuencias, para con ello determinar los requerimientos de Alta Disponibilidad que se requieran.

Diseño del Sistema de Alta Disponibilidad: Una vez hecho el análisis, el siguiente paso será la conjunción de los mecanismos y técnicas tratándolos como elementos de un sistema, es decir, diseñar un sistema que solucione la problemática a tratar. Se determinarán entre otros elementos, el porcentaje de disponibilidad anual y el nivel de disponibilidad del servicio.

Desarrollo del Sistema de Alta Disponibilidad: Consistirá básicamente en la Eliminación de los Puntos Simples de Fallo y en la programación del *software* de Alta Disponibilidad empleando los lenguajes de programación y las herramientas que sean necesarias para su desarrollo.

Implementación del Sistema de Alta Disponibilidad: Una vez probado y liberado el Sistema de Alta Disponibilidad, visto en este paso ya como una herramienta, será implementada de acuerdo a las especificaciones hechas en el análisis, además de complementar dicha implementación con la elaboración de procedimientos, uno de ellos es el Plan de Contingencias.

Conclusiones: Reflexión sobre si se cumplió o no con el objetivo, cuales son las limitantes y los factores críticos de éxito para el Sistema de Alta Disponibilidad, las contribuciones hechas por el presente trabajo y por último las líneas futuras que se abren a partir de dicho trabajo.

CAPÍTULO 1

Antecedentes y Marco Teórico

CAPÍTULO 1: ANTECEDENTES Y MARCO TEÓRICO

1.1 ¿QUÉ ES ALTA DISPONIBILIDAD?

1.1.1 DEFINICIONES

Para entender este concepto aplicado a las Tecnologías de la Información, podemos empezar por definir algunos términos como son "Disponibilidad", "Alta Disponibilidad", y "Alta Disponibilidad Computacional".

El término *Disponibilidad* describe un sistema que proporciona un nivel específico de servicio como necesario. En Tecnologías de la Información, Disponibilidad es generalmente entendido como el periodo de tiempo en el cual los servicios están activos o como el tiempo requerido para que el sistema responda a los usuarios.

El término A*lta Disponibilidad* caracteriza a un sistema que es diseñado para evitar la pérdida del servicio por reducción o administración de fallos así como la minimización de las interrupciones planeadas. Esperamos que un servicio sea altamente disponible cuando la vida, la salud, el bienestar e incluyendo el aspecto económico dependa de ello.

Por ejemplo, esperamos que los servicios de electrificación sean altamente disponibles, de manera que cualquier paro por muy corto que éste sea resulta inaceptable debido a que muchos aspectos de nuestra vida giran en torno a este servicio. Incluso hasta el servicio más altamente disponible tiene paros ocasionales, todos hemos experimentado alguna vez un corte de corriente incluso en una gran ciudad. Para estos casos, esperamos ver un restablecimiento del servicio de inmediato. Cuando una de estas fallas ocurre, esperamos que la compañía que administra el servicio esté en camino para arreglar el problema lo más pronto posible.

En muchos negocios, la disponibilidad de las computadoras ha tomado tanta importancia como la disponibilidad de la energía eléctrica. La *Alta Disponibilidad Computacional* emplea sistemas de cómputo los cuales son diseñados y administrados para operar con sólo una mínima cantidad de interrupciones planeadas y no planeadas.

1.1.2 CONSIDERACIONES

La *Alta Disponibilidad no es absoluta*. Varía en función de las necesidades de las empresas. Algunas empresas o compañías internacionales realizan múltiples transacciones que requieren acceso a bases de datos todo el tiempo. Instituciones financieras hacen transacciones en cualquier momento del día o de la noche, siete días a la semana. Por otra parte, medianas empresas podrían requerir que la

computadora esté disponible solo 18 horas al día, pero durante esas 18 horas ellos podrían requerir Alta Disponibilidad.

Alta Disponibilidad no implica Disponibilidad Continua. La Disponibilidad Continua significa que un servicio nunca se deja de proporcionar, no existen interrupciones planeadas o no planeadas, simplemente no hay interrupciones, por lo que podría considerarse entonces que ésta es la meta más ambiciosa de la Alta Disponibilidad, donde no hay un lapso de inactividad en el servicio. La Disponibilidad Continua es un estado ideal, sin embargo, este término es usado frecuentemente para indicar un muy alto nivel de disponibilidad en el cual sólo pequeños intervalos de inactividad son aceptables.

Alta Disponibilidad no implica Tolerancia a Fallos. La Tolerancia a Fallos es un método para lograr muy altos niveles de disponibilidad. Un Sistema Tolerante a Fallos se caracteriza por la redundancia de la mayoría de sus componentes *hardware*: CPU, memoria, subsistemas de entrada y/o salida y otros componentes. Un Sistema Tolerante a Fallos tiene la habilidad de continuar con el servicio ante una falla de *hardware* o *software*, pero no está exento de interrupciones no planeadas provocadas por errores humanos.

La Alta Disponibilidad deberá ser la adecuada a las necesidades de los usuarios. La sensibilidad del usuario ante interrupciones no planeadas del servicio depende de la aplicación en específico. Por ejemplo, un fallo corregido después de un segundo podría no ser considerado como una interrupción no planeada del servicio en un ambiente que realiza procesos transaccionales en línea; sin embargo, para una aplicación científica que es ejecutada en tiempo real, un segundo puede considerarse un intervalo inaceptable.

Por tanto, el desafío es diseñar sistemas en los cuales los problemas puedan ser predecidos y aislados antes de que un fallo ocurra, y los fallos detectados y corregidos cuando aparezcan.

1.1.3 ALTA DISPONIBILIDAD COMO NIVEL DE SERVICIO

El nivel de servicio de un sistema es el grado de servicio que el sistema proporcionará a sus usuarios. El nivel de servicio está especificado por un estándar conocido como *Service Level Agreement* (SLA), el cual sirve para garantizar la calidad de un servicio. El nivel de servicio que un negocio requiere es determinado por la clase de aplicaciones que desarrolla, y los Sistemas de Alta Disponibilidad proporcionan la estructura *hardware-software* en la cual estas aplicaciones puedan trabajar efectivamente para proporcionar el nivel de servicio necesario. La Alta Disponibilidad implica un nivel de servicio en el cual, interrupciones planeadas y/o no planeadas, no excedan la cota mínima establecida.

1.1.4 ALTA DISPONIBILIDAD COMO REQUERIMIENTO

En la actualidad la Alta Disponibilidad Computacional es ya considerada un requerimiento y no un lujo. Dentro del mundo de los negocios, podemos ver la Alta Disponibilidad desde dos perspectivas, como un seguro contra la pérdida de negocios debido a una interrupción en los sistemas; y como aquella que proporciona nuevas oportunidades que permitan a la compañía proveer servicios mejores y más competitivos a sus consumidores.

La Alta Disponibilidad Computacional es frecuentemente comprendida como un seguro contra la pérdida de entradas, la insatisfacción del cliente y la pérdida de oportunidades. Una solución de Alta Disponibilidad es necesaria cuando la pérdida del sistema reditúa en pérdida de ingresos. Muchas veces, la aplicación se dice que debe ser de misión crítica. Para todas las aplicaciones de misión crítica, la Alta Disponibilidad es un requerimiento. En la banca, por ejemplo, la habilidad de obtener ciertos balances de cuenta las 24 horas al día puede ser de misión crítica.

Sistemas altamente disponibles proporcionan una oportunidad de negocio, desde que existe una demanda creciente de servicios computarizados en áreas como la banca, mercados financieros, comunicaciones, ventas en línea, etc.

No es posible dar una simple definición de cuándo una aplicación es de misión crítica o cuándo la Alta Disponibilidad de las aplicaciones crea nuevas oportunidades, esto depende de la naturaleza de la empresa. Sin embargo, para cualquier empresa que dependa de sistemas computacionales, los siguientes principios son siempre verdaderos (Ver Tabla 1.1):

El nivel de disponibilidad requerido es determinado por las necesidades de la empresa. No hay una cantidad absoluta de disponibilidad que sea correcta para todas las empresas
Hay muchas formas de proporcionar Alta Disponibilidad
La Alta Disponibilidad afecta a todos los aspectos del sistema
La probabilidad de fallas puede ser reducida creando una infraestructura que de énfasis a los procedimientos claros y al mantenimiento preventivo
La recuperación ante fallas deberá ser planificada

Tabla 1.1 Principios sobre Alta Disponibilidad

Lo que se espera de las aplicaciones que se ejecutan en ambientes de misión crítica (Ver Tabla 1.2):

Debería existir una tasa baja de fallas en la aplicación, esto es, un máximo tiempo entre fallas
Las aplicaciones debieran poder ser recuperadas después de fallo
Debiera de ser minimizado el número de interrupciones planeadas
El sistema debiera ser configurable sin apagarlo
Las herramientas de administración del sistema debieran estar disponibles

Tabla 1.2 Comportamiento de las aplicaciones

1.1.5 EL COSTO DE LA ALTA DISPONIBILIDAD

El costo de la Alta Disponibilidad depende en gran medida del nivel de disponibilidad que se elija. De esta forma el valor de la Alta Disponibilidad en la empresa está directamente relacionado con los costos provocados por las interrupciones. Un alto costo de interrupciones, es la forma más fácil de justificar una costosa solución de Alta Disponibilidad.

Como el nivel de disponibilidad está aproximado al ideal 100% de disponibilidad, el costo de la solución incrementa rápidamente. De está forma, el costo de 99.95% de disponibilidad es significativamente mayor que el costo de 99.5% de disponibilidad, y el costo del 99.5% de disponibilidad es significativamente mayor que el 99% de disponibilidad y así sucesivamente.

1.2 INTERRUPCIÓN DEL SERVICIO

1.2.1 DEFINICIÓN Y CLASIFICACIÓN

Si hablamos de Alta Disponibilidad, obligadamente tenemos que hablar de sus inconvenientes u obstáculos. En el argot computacional están definidos formalmente los siguientes términos:

Fallo (failure): Se define como la suspensión de la operación normal de algún componente, los fallos ocurren a nivel *hardware*, *software*, administración de sistema y red, inclusive en el ambiente.

Paro (outage): Se define como la pérdida específica de un servicio que es percibido por el usuario. No todos los fallos causan paros y no todos los paros no planeados son causados por fallos.

Interrupción o tiempo fuera de servicio (downtime): Se define como la duración de un paro del servicio.

Las interrupciones se clasifican en dos rubros:

Interrupciones planeadas: Las que se realizan cuando paramos el sistema para realizar cambios o mejoras en el *hardware* y/o *software* que compone al sistema. Generalmente este tipo de interrupciones son necesarias e involucran procesos de actualización y mantenimiento físico y/o lógico de los sistemas por parte de los administradores.

Interrupciones no planeadas: Las que suceden por acontecimientos imprevistos (como un apagón, un error del *hardware* o del *software*, problemas de seguridad, un desastre natural, virus, accidentes, caídas involuntarias del sistema).

En ambos casos, durante este tiempo de no operatividad, la suspensión de las operaciones del sistema afecta a la empresa o institución no tan solo desde el punto de vista de ingresos económicos, sino también a su imagen.

Aún así, existen empresas que pueden permitirse estar no operativas durante horas, aunque existen sectores de negocios que no pueden permitirse la más mínima interrupción (banca, hospitales, logística, transporte, comercio electrónico, etc.).

Para aquellas empresas que se pueden permitir estar no operativas durante horas, es recomendable contar con mecanismos como son los Planes de Recuperación o Contingencia que garanticen que las pérdidas de datos y consecuentemente el costo de la interrupción será mínimo. Incluyendo la creación de copias de seguridad de archivos, seguridad, configuraciones, programas y sus fuentes, por lo menos diariamente. Mantener las copias de seguridad en un lugar seguro y de ser posible en una ubicación distinta al del sistema en producción

Para el caso de las interrupciones planeadas, la meta será minimizar su número, sólo cuando sea obligadamente necesario se deberían de hacer.

Para el caso de las interrupciones no planeadas, la meta será predecir y aislar los problemas que las ocasionen, así como detectarlas y corregirlas inmediatamente cuando aparezcan.

Es decir, las interrupciones pueden producirse por distintos motivos, lo que nos obliga a estar preparados para que cualquier eventualidad sea corregida en el menor tiempo posible y reducir al máximo las pérdidas que éstas nos puedan acarrear.

1.2.2 EL SIGNIFICADO DE UNA INTERRUPCIÓN EN EL TIEMPO

Un aspecto importante es el tiempo en el cual el sistema estará fuera de servicio. Como ya lo he hecho notar, dependiendo de la aplicación, el tiempo que dure la interrupción del servicio podría ser significante o no. Una interrupción de 10 segundos podría no ser considerada crítica, pero dos horas podrían ser fatales para una aplicación, mientras que otra aplicación no toleraría más de 10 segundos de inactividad.

La importancia de la Alta Disponibilidad puede observarse con el siguiente escenario. Un sistema computacional sufre una interrupción de sus servicios debido a que un disco duro sufrió un fallo. La Figura 1.1 muestra una secuencia de eventos que podrían llevarse a cabo cuando un cliente experimenta un fallo en su disco duro en un sistema convencional que no usa discos de datos replicados.

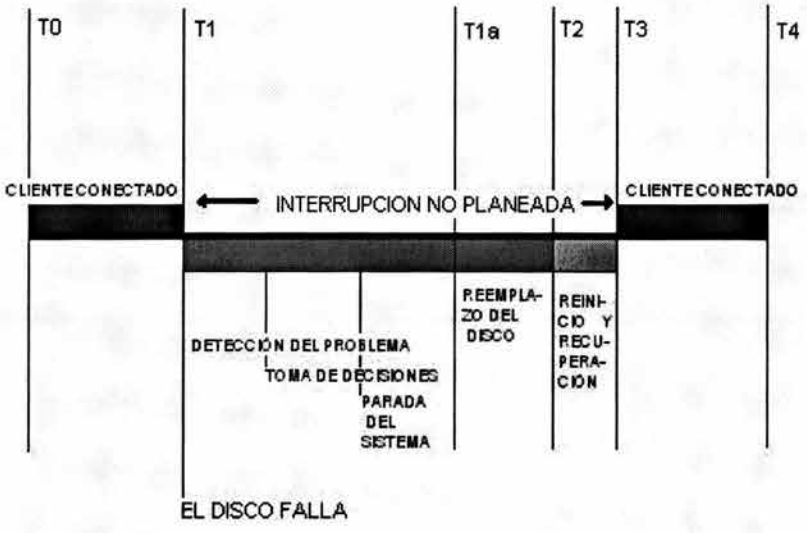

Figura 1.1 Línea del tiempo de una interrupción en un sistema convencional

Cuando el disco falla, el ambiente operativo del sistema queda disponible hasta que el disco sea reemplazado. El fallo toma lugar en T1, y la transacción del usuario es abortada. El sistema se reactiva hasta T3, precedido de un reemplazo del disco dañado, un reinicio del sistema y la recuperación de la base de datos. Incluyendo la restauración de los datos a partir de una copia de seguridad. Esta secuencia puede suceder durante unas pocas horas o todo un día. En este escenario, el tiempo para recuperar el sistema es totalmente impredecible.

La Figura 1.2 muestra el mismo escenario cuando el sistema usa una característica de la Alta Disponibilidad conocida como *disk mirroring* o discos replicados, lo cual previene la pérdida del servicio.

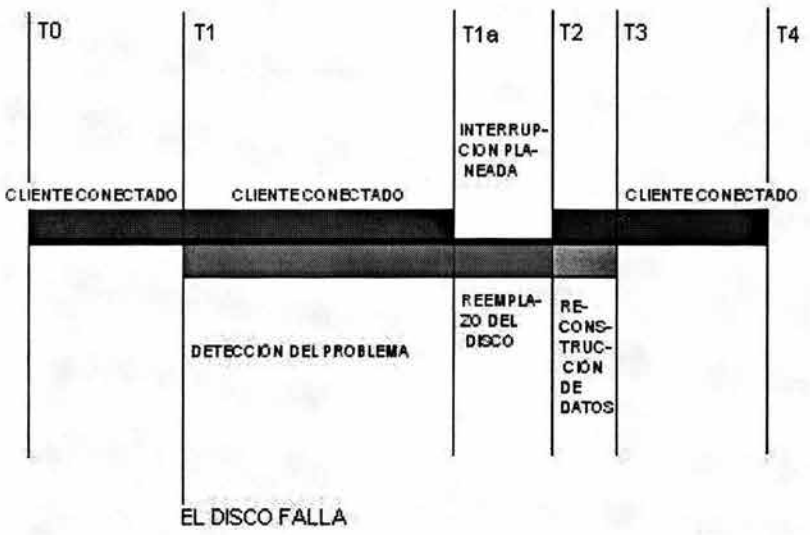

Figura 1.2 Línea del tiempo de una interrupción en un sistema que emplea discos replicados

Cuando el disco falla, el disco replicado continúa disponible, así que no hay pérdida de datos, y el servicio continúa. Además, el reemplazo del disco dañado puede ser aplazado a un periodo de mantenimiento. Una diferencia significativa entre este escenario y el anterior es que se puede predecir la cantidad de tiempo necesario para reparar. Con el uso de discos replicados, una cantidad impredecible de tiempo fuera de servicio es reemplazado por una pequeña interrupción planeada.

1.2.3 CAUSAS DE LAS INTERRUPCIONES PLANEADAS

Las interrupciones planeadas del servicio se incluyen en una aplicación para llevar a cabo una copia de seguridad previamente planificada o actualizar el *software*. Algunas otras son las listadas en la Tabla 1.3:

Realización de copias de seguridad o respaldos de manera periódica
Actualización del *software*
Expansión o reparación de *hardware*
Cambios en la configuración del sistema
Cambios en los bancos de datos

Tabla 1.3 Algunas causas de las interrupciones planeadas

Estas interrupciones normalmente no causan problemas si cada una de ellas es apropiadamente planificada. Algunos ambientes de procesamiento de datos pueden tolerar pequeñas interrupciones planeadas, otros pueden tolerar periodos regulares de paro del servicio cada día o semana.

Una alternativa a las interrupciones planeadas son las operaciones de mantenimiento mientras el sistema está en línea. Las operaciones de respaldo que se llevan acabo mientras el sistema está ejecutándose son conocidas como *on-line backups* (respaldos en línea). Las actualizaciones o reparaciones de *hardware* mientras el sistema está ejecutándose son conocidas como *hot plug operations* (operaciones en caliente).

1.2.4 CAUSAS DE LAS INTERRUPCIONES NO PLANEADAS

Los elementos de la Tabla 1.4 son algunas de las causas que producen interrupciones no planeadas:

Falla en el *hardware*
Sistema de archivos a su máxima capacidad
Fallas en el *kernel* del sistema operativo
Falla del respaldo
Dispositivos de almacenamiento interno a su máxima capacidad
Picos de energía
Falla en la energía
Problemas en la infraestructura LAN
Defectos en el *software*
Falla de las aplicaciones
Defectos en el *firmware*
Desastres naturales (incendios, inundaciones, etc.)
Errores del administrador u operador

Tabla 1.4 Algunas causas de la interrupciones no planeadas

Una interrupción no planeada produce un impacto muy negativo. Los efectos de las interrupciones no planeadas incluyen la espera en línea de los usuarios durante la caída del sistema, que los aviones no puedan partir o planear porque el control de tráfico aéreo falló, una línea de ensamblado tiene que parar, los doctores no pueden obtener los datos de sus pacientes del sistema de información del hospital, etc. En muchos casos, los negocios se pierden porque las transacciones no pueden ser completadas. Una interrupción no planeada en la mayoría de los casos reduce la satisfacción del usuario.

Analicemos la situación partiendo de un estudio hecho en 1993 por *Gartner Group*, el cual mide las causas que producen las interrupciones no planeadas del servicio (Ver Figura 1.3).

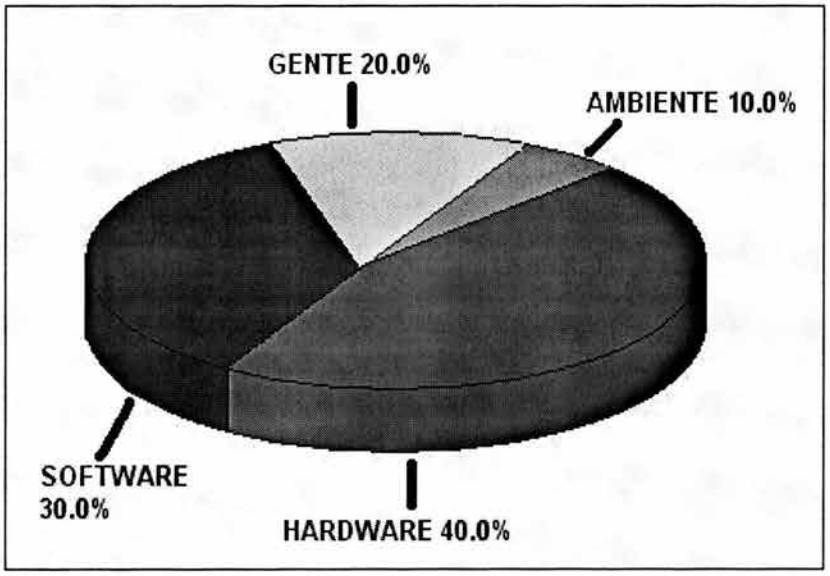

Figura 1.3 Causas de las interrupciones no planeadas. Estudio hecho en 1993 por *Gartner Group*

Tradicionalmente, la mayoría de la gente piensa solo en el *hardware* cuando empiezan a analizar sus requerimientos de Alta Disponibilidad. Sin embargo, como lo muestra el estudio, existen más factores que deben ser considerados cuando se trate de diseñar e implantar un verdadero ambiente de Alta Disponibilidad. Las categorías de *software*, gente (administradores de sistema y red), y elementos externos como la electrificación y los servicios de telecomunicación, clima, y tiempo deben ser propiamente consideradas.

La Figura 1.4 muestra las principales causas de las interrupciones no planeadas, de acuerdo a un estudio hecho en 2001 por *Gartner/Dataquest.*

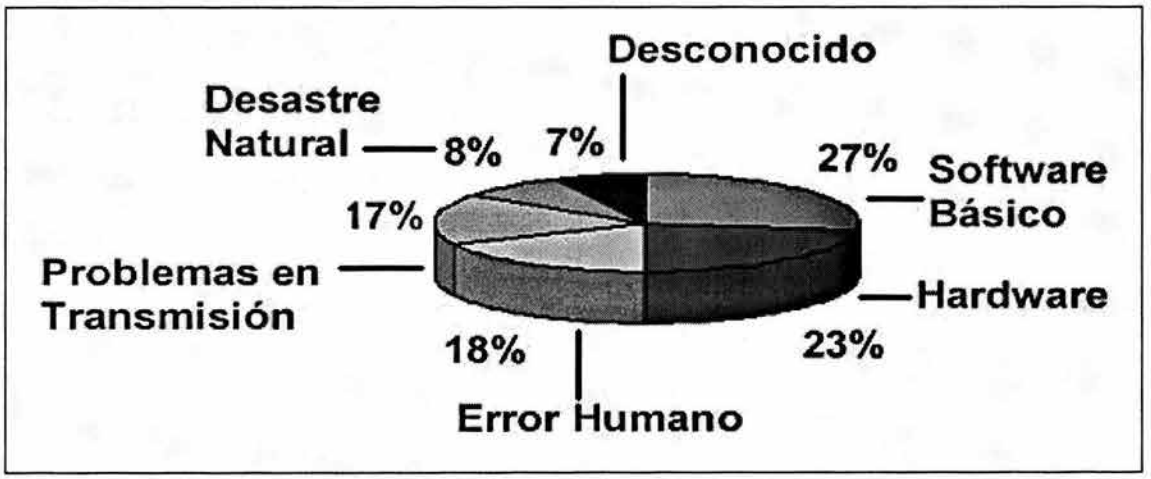

Figura 1.4 Causas de las interrupciones no planeadas. Estudio hecho en 2001 por *Gartner/Dataquest*

Podemos observar que en este estudio se consideran más aspectos que en el estudio previo, e incluso los fallos en el *hardware*, como causa de interrupciones no planeadas, se han minimizado y ahora los fallos por el *software* son causa de más peso.

La figura 1.5 muestra otra perspectiva mucho más particular sobre las causas de interrupciones no planeadas, según un estudio hecho por *IBM* en 2002:

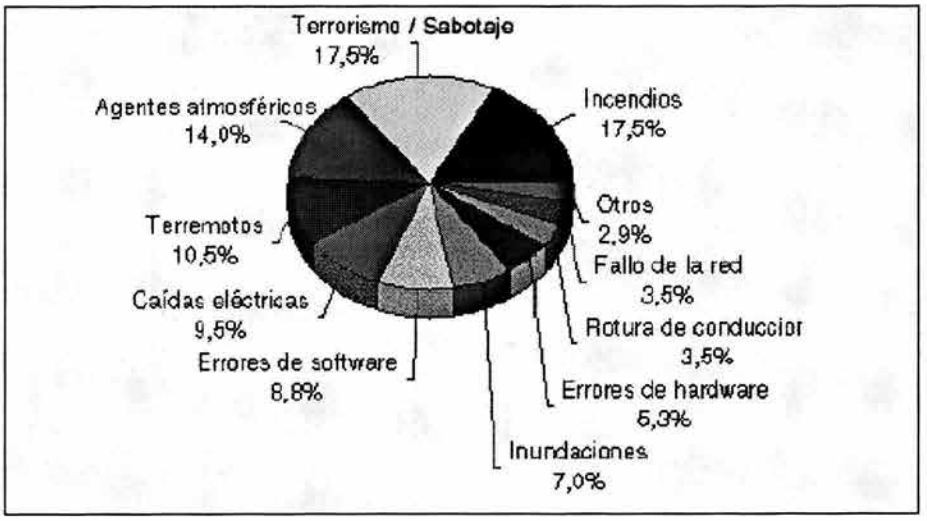

Figura 1.5 Causas de las interrupciones no planeadas. Estudio hecho en 2002 por *IBM*

Podemos observar que las causas se han diversificado más, y aparecen nuevos aspectos como son el terrorismo y el sabotaje que encabezan la lista. Obviamente este aspecto nada tiene que ver con la infraestructura computacional de una empresa o institución.

1.2.5 CONSECUENCIAS DE LA FALTA DE DISPONIBILIDAD DEL SERVICIO

En 2001, la compañía *IBM* realizó un estudio sobre la disponibilidad de los servicios que proporcionan las empresas o instituciones, e hizo una clasificación por sectores, en la Figura 1.6 se muestra cuáles sectores y en que porcentaje sufren más interrupciones:

Figura 1.6 Sectores que más sufren de interrupciones del servicio

Podemos observar que Banca y Finanzas, Gobierno, Administraciones Públicas e Instituciones y Educación son los sectores que más interrupciones sufren, por tanto es importante analizar las consecuencias de dichas interrupciones.

Entre todas las posibles consecuencias que genera la falta de disponibilidad de los servicios que proporcione una empresa o institución tenemos los listados en la Tabla 1.5 que se muestra a continuación:

Pérdida de Clientes
Pérdida de oportunidades
Pérdida de capacidad
Trabajo perdido o improductivo
Costos de restauración
Penalizaciones
Mala publicidad
Pérdida de ingresos

Tabla 1.5 Posibles consecuencias que origina la falta de disponibilidad

En la Tabla 1.6 podemos observar la pérdida de ingresos (en dólares) por hora que sufren algunos sectores dada una interrupción de los servicios que proporciona:

SECTOR	PÉRDIDA DE INGRESOS (Dls.) POR HORA
Energía	2,817,846
Telecomunicaciones	2,066,245
Financiero	1,495,134
Comercio	1,107,274
Químico	704,101
Salud	636,030
Entretenimiento	340,432

Tabla 1.6 Pérdida de ingresos por hora de algunos sectores. Fuente: *Network Computing*, marzo de 2001

El cálculo de los costos de las interrupciones requiere estimar las pérdidas de los ingresos, los costos del personal y los costos intangibles, causados por un fallo del sistema y sumar los tres rubros.

Paso 1. Calcular las pérdidas de ingresos: Existe un método simple y otro método más complejo para calcular las pérdidas. La fórmula simple calcula primero el ingreso por una hora de funcionamiento del negocio (Ver Fórmula 1.1):

$$INGRESOS = \frac{INGRESO\_ANUAL\_TOTAL}{HORAS\_HÁBILES\_AL\_AÑO}$$

Fórmula 1.1 Cálculo de los ingresos por hora hábil

Donde los ingresos son calculados en unidades monetarias (UM) por hora y las horas hábiles al año se calculan con la Fórmula 1.2:

$$HORAS\_HÁBILES\_AL\_AÑO = HORAS\_HÁBILES * DÍAS\_HÁBILES * MESES\_HÁBILES$$

Fórmula 1.2 Cálculo de las horas hábiles al año

Después, para obtener la cifra de perdida de ingresos, multiplicamos los ingresos por hora por el número de horas de interrupción.

Ejemplo 1.1: Un negocio funciona de 8 a.m. a 8 p.m., de lunes a sábado, todas las semanas del año. Los ingresos por ventas del año pasado fueron de 6,000 millones de UM. Si procesamos estos datos con nuestra ecuación obtenemos:

$$INGRESOS = \frac{6{,}000{,}000{,}000UM}{12 * 22 * 12HORAS} = 1{,}893{,}939\frac{UM}{HORA}$$

El ingreso por hora asciende a 1,893,939 UM. De acuerdo con este cálculo, una incidencia de 12 horas de interrupción, implica una pérdida de ingresos que asciende a 22,727,268 UM (12 horas por 1,893,939 UM).

El considerar que el negocio no genera ingresos, mientras el sistema no esté disponible, es indudablemente una suposición importante. Pero como mínimo, este enfoque nos da un punto de partida y unos datos estadísticos, para comprender el valor de cada hora disponible de la organización. Si parece que el método simple no refleja la realidad de una manera suficientemente precisa, puede adoptar el siguiente enfoque (Ver Fórmula 1.3):

$$PÉRDIDA\_DE\_INGRESOS = UMPD + UMPAN + UMPAP$$

Fórmula 1.3 Cálculo de la pérdida de ingresos

Donde:

UMPD	=	UM de Pedidos desviados
UMPAN	=	UM de Pedidos anulados
UMPAP	=	UM de Pedidos aplazados

Pedidos desviados son aquellos que nuestros clientes han hecho a nuestros competidores, porque nuestra empresa no ha podido responder. Pedidos anulados son los pedidos que los clientes ya habían hecho, pero posteriormente los anularon, al no poder aceptar el retraso en su petición. Las UM de pedidos aplazados cuantifican los efectos de los retrasos sobre el flujo de caja y otras posibles pérdidas de rendimiento.

Paso 2. Cálculo de costos de personal. Otro factor a considerar, cuando calculamos los costos de las interrupciones, son los costos de personal generados mientras el sistema no estaba disponible. Para ello emplearemos la Fórmula 1.4 presentada a continuación, para calcular la pérdida laboral:

$$PÉRDIDA\_LABORAL = No.PERSONAS * \%HA * No.HORAS * CEH$$

Fórmula 1.4 Cálculo de la pérdida laboral

Donde:

%HA	=	Porcentaje de horas afectado
CEH	=	Es el costo de un empleado por hora, definido por la Fórmula 1.5

$$CEH = PAGA\_HORA + BENEFICIOS + GASTOS\_FIJOS$$

Fórmula 1.5 Cálculo del Costo de un empleado por hora

Ejemplo 1.2: Supongamos que en la empresa, el costo total promedio por empleado por hora es de 2,500 UM. Si 50 empleados son afectados en un 50% durante las primeras 2 horas del fallo del sistema, 100 personas se ven afectadas en un 75% por las próximas 2 horas y 200 personas pierden todo su tiempo (100%), cuando el fallo dura más de 4 horas, un tiempo muerto de 12 horas costaría a la empresa 4,500,000 UM. Dividiendo este costo total de personal, entre el número de horas que duró el incidente (12), obtenemos el costo promedio por hora: 375,000 UM. Esto quiere decir que cada hora de fallo del sistema le cuesta a la empresa un promedio de 375,000 UM en personal.

Paso 3. El cálculo de costos intangibles. Por más intangibles que sean, estos costos son algo que tiene que ser tomado en consideración, porque ponen de manifiesto algunos de los efectos a largo plazo de los fallos de disponibilidad de sistemas. Mencionaré sólo algunas cosas en las que hay que pensar.

Pérdida de reputación: ¿Quizás los clientes no sean tan fieles, después de una incidencia que impidió a la empresa en cuestión satisfacerles durante un período de tiempo? ¿Es ésta la oportunidad para ellos, para "salir de compras" por ahí?

El efecto secundario sobre los clientes de los clientes de la empresa. ¿Hay otras empresas o personas, aparte de los clientes primarios que pudieran ser afectadas?

La imagen. Si un cliente ha tenido una mala experiencia se lo dice a otros. Esto puede extender y amplificar el daño a la reputación de la empresa.

El ánimo dentro de la empresa. Si los empleados no pueden cumplir con su trabajo, debido a fallos del sistema, su ánimo afecta su modo de trabajar y prestar servicios.

La pérdida de ingresos, los costos perdidos de personal y los costos intangibles se suman, para obtener el costo por hora de la no disponibilidad del sistema.

Aunque los efectos de las interrupciones de sistemas no se pueden medir con facilidad, es muy necesario realizar estos cálculos de costos de los fallos de disponibilidad. El alcance de los efectos de un fallo, no depende solamente de su duración. Su cálculo supone también la evaluación de si, durante todo el tiempo

del fallo del sistema, los datos realmente necesitaban ser disponibles. Muchas interrupciones planeadas, aprovechan el hecho que la duración de la falta de disponibilidad, muchas veces, no es tan importante como el cuándo el sistema no está disponible.

1.3 MÉTODOS PARA CUANTIFICAR LA DISPONIBILIDAD DE UN SISTEMA

Existen dos métodos para cuantificar la disponibilidad de un sistema informático, los cuales se describen a continuación, sin embargo ambos nos llevan a las mismas conclusiones, así que es indistinto si usamos una fórmula o la otra.

1.3.1 PRIMER MÉTODO

Esta métrica es la más comúnmente utilizada para medir la disponibilidad y fiabilidad de un sistema, se basa en el tiempo promedio entre fallos o MTBF (*Mean Time Between Failure*) que mide el tiempo promedio transcurrido hasta que un dispositivo falla, y el tiempo promedio de recuperación o MTTR (*Mean Time To Recover*) que mide el tiempo promedio tomado en restablecer la situación normal una vez que se ha producido el fallo.

La Fórmula 1.6 permite calcular la disponibilidad de un sistema informático:

$$DISPONIBILIDAD = \frac{MTBF}{MTBF + MTTR}$$

Fórmula 1.6 Cálculo de la disponibilidad de un sistema informático (I)

El tiempo promedio entre fallos se calcula con la Fórmula 1.7:

$$MTBF = \frac{TOT}{TNOF}$$

Fórmula 1.7 Cálculo del tiempo promedio entre fallos

Donde:
MTBF = *Mean Time Between Failure* (Tiempo Promedio entre Fallos)
MTTR = *Maximum Time To Repair* (Máximo Tiempo de Reparación)
TOT = *Total Operating Time* (Tiempo Total de Operación)
TNOF = *Total No. Of Failures* (Número Total de Fallos o Interrupciones)

Ejemplo 1.3: Pensemos en una empresa que requiere que sus sistemas estén en disponibilidad las 24 horas de cada día, los 7 días de la semana y los 365 días del año (sistemas 24x7x365), por tanto el total de horas es de 8,760. Dada la experiencia y los datos que arrojan las estadísticas, se piensa que durante el año habrá 5 interrupciones (planeadas o no planeadas) del servicio aproximadamente. Además, se piensa que tardarán alrededor de 2 horas para restablecer el servicio.

Con estos datos, podemos calcular la disponibilidad de sus sistemas:

$$MTBF = \frac{8760 HORAS}{5 INTERRUPCIONES} = 1752 HORAS$$

Y $$MTTR = 2 HORAS$$

Por lo tanto:

$$DISPONIBILIDAD = \frac{1752 HORAS}{1752 HORAS + 2 HORAS} = 0.99885974914481185860889395667047$$

En porcentaje y aplicando un redondeo a tres cifras decimales:

$$\%DISPONIBILIDAD = 99.886$$

1.3.2 SEGUNDO MÉTODO

La Fórmula 1.8 define la disponibilidad como un porcentaje del tiempo que una unidad puede ser usada. Este tiempo es continuo, la suma del tiempo de operación y el tiempo en el cual la unidad está fuera de servicio.

$$\%DISPONIBILIDAD = \frac{TET - SOIT}{TET} X100$$

Fórmula 1.8 Cálculo de la disponibilidad de un sistema informático (II)

TET = *Total Elapsed Time* (Total del Tiempo Transcurrido)
SOIT = *Sum Of Inoperative Times* (Suma de los Tiempos de Inoperatividad)

Aplicando este método al Ejemplo 1.3 tenemos que:

$$TET = 7860 HORAS$$
$$SOIT = (5 INTERRUPCIONES) x (2 HORAS) = 10 HORAS$$

$$\%DISPONIBILIDAD = \frac{7860 HORAS - 10 HORAS}{7860 HORAS} X100 = 99.872773536895674300254452926209$$

Aplicando redondeo a tres cifras decimales:

$$\%DISPONIBILIDAD = 99.873$$

1.3.3 PERIODOS ESPERADOS DE OPERACIÓN

La medida de la disponibilidad debe ser comparada contra el periodo esperado de operación deseado por la empresa, de acuerdo a sus requerimientos e intereses. Las siguientes Tablas muestran los *uptime* y *downtime* (tiempo de operatividad y tiempo de inoperatividad, respectivamente) asociados con los diferentes porcentajes de disponibilidad para dos periodos comunes de operación. Con relación a esto, la Tabla 1.7 muestra un sistema con periodo de operación 24x7x365, el cual se espera que sea usado las 24 horas del día, los 7 días de la semana, 365 días del año.

DISPONIBILIDAD	UPTIME MÍNIMO ESPERADO	DOWNTIME MÁXIMO PERMITIDO	TIEMPO RESTANTE
99%	8,672	88	0
99.5%	8,716	44	0
99.95%	8,755	5	0
100%	8,760	0	0

Tabla 1.7 Porcentajes de disponibilidad para un sistema 24x7x365

Asimismo la Tabla 1.7 muestra que no existe tiempo restante en un sistema 24x7x365, todo el tiempo disponible en el año es contabilizado; esto significa que todo el mantenimiento debe ser realizado cuando el sistema está activo o durante el *downtime* máximo permitido. Además, entre más alto es el porcentaje de disponibilidad, el tiempo permitido para fallar es menor. La Tabla 1.8 muestra un sistema 12x5x52, del cual se espera este disponible 12 horas del día, 5 días de la semana, 52 semanas del año.

DISPONIBILIDAD	UPTIME MÍNIMO ESPERADO	DOWNTIME MÁXIMO PERMITIDO	TIEMPO RESTANTE
99%	3,008	32	5,642
99.5%	3,104	16	5,642
99.95%	3,118	2	5,642
100%	3,118	0	5,642

Tabla 1.8 Porcentajes de disponibilidad para un sistema 12x5x52

La Tabla 1.8 muestra que para un sistema 12x5x52, hay 5,642 horas de tiempo restante, el cual puede ser usado para operaciones de mantenimiento planificado, donde el sistema pudiera ser interrumpido.

Por otra parte, la Tabla 1.9 muestra datos relacionados a un sistema 24x7x365 que son más conocidos y difundidos por analistas y diseñadores de Sistemas de Alta Disponibilidad. En los Estados Unidos de Norteamérica es muy común escuchar la frase "cinco nueves de disponibilidad" refiriéndose a 99.999% de disponibilidad de un sistema. Por ejemplo, comercialmente es más atractivo escuchar que las soluciones existentes en el mercado garantizan 99.999%, 99.99% ó 99.9% de disponibilidad. Entre estos porcentajes existe un pequeñísimo abismo, reflejado en más recursos y por ende la solución es más costosa.

%UPTIME	%DOWNTIME	TIEMPO SIN SERVICIO ANUAL	NORMALIZADO (SEGUNDOS)
98	2	7.30 días	630,720
99	1	3.65 días	315,360
99.8	0.2	17 horas, 30 minutos	63,000
99.9	0.1	8 horas, 45 minutos	31,500
99.99	0.01	52 minutos, 30 segundos	3,150
99.999	0.001	5 minutos, 15 segundos	315
99.9999	0.0001	31.5 segundos	31.5

Tabla 1.9 Porcentajes de disponibilidad comerciales para un sistema 24x7x365

1.4 NIVELES DE DISPONIBILIDAD

Con base en el porcentaje de disponibilidad y los requerimientos e intereses de la empresa, podemos ubicar el nivel de disponibilidad que requieren los sistemas informáticos. Los niveles de disponibilidad permiten tanto al usuario final y al analista-diseñador de Sistemas de Alta Disponibilidad, tener una idea objetiva sobre la disponibilidad que tendrán los servicios una vez que sean implantados, cada nivel especifica, con base en las funciones de la empresa o institución y el porcentaje de disponibilidad requerido, el tipo de sistema de Alta Disponibilidad a diseñar y los mecanismos básicos a implantar.

La Tabla 1.10 describe los niveles de disponibilidad:

NIVEL	FUNCIONES	PORCENTAJE DE DISPONIBILIDAD	SISTEMA DE ALTA DISPONIBILIDAD SUGERIDO	MECANISMOS BÁSICOS A IMPLANTAR
Convencional	Las funciones pueden verse interrumpidas y la integridad de los datos no es esencial	Menor al 90%	Sistema de la Disponibilidad Base: El sistema está listo para el uso inmediato, pero experimentará tanto interrupciones planeadas como no planeadas	Respaldo tradicional
Media	Las funciones pueden verse interrumpidas y la integridad de los datos no es esencial	Menor al 95%	Los sistemas de la Disponibilidad Base: El sistema está listo para el uso inmediato, pero experimentará tanto interrupciones planeadas como no planeadas	*Journaling*
Alta Disponibilidad	Las funciones aceptan pequeñas interrupciones y al retomar se reprocesan transacciones	Menor al 99%	Sistema de Disponibilidad Alta: Incluyen tecnologías para reducir drásticamente el número y la duración de interrupciones imprevistas. Todavía existen interrupciones planeadas, pero los servidores incluyen herramientas que reducen su impacto	*Journaling, Recovery* automático

Tabla 1.10 Niveles de disponibilidad

NIVEL	FUNCIONES	PORCENTAJE DE DISPONIBILIDAD	SISTEMA DE ALTA DISPONIBILIDAD SUGERIDO	MECANISMOS BÁSICOS A IMPLANTAR
Resistencia a Fallos	Requiere de operación sin interrupción en horario laboral, la retoma en caso de falla es automática	Menor al 99.9%	Entornos de operaciones continuas: Utilizan tecnologías especiales para asegurarse de que no hay interrupciones planeadas para respaldos, actualizaciones, u otras que obliguen a no tener el sistema disponible	*Clustering, mirroring*
Tolerancia a Fallos	Capacidad de procesamiento continuo y cualquier fallo debe ser transparente para el usuario	Menor al 99.99%	Sistema Tolerante a Fallos: Va un paso más lejos para asegurarse de que no habrán interrupciones en los sistemas. Para alcanzar este nivel de disponibilidad, se deben utilizar servidores duales o los clusters de servidores redundantes donde un servidor asume el control automáticamente si el otro servidor cae	Duplicidad y redundancia
Tolerancia a Desastres	Disponibilidad en todo momento, capacidad para soportar desastres naturales y humanos	Menor al 99.999%	Sistema Tolerante a Desastres: Requieren de sistemas alejados entre sí para asumir el control en cuanto pueda producirse una interrupción provocada por un desastre	Duplicidad, redundancia y mecanismos de *recovery*

Tabla 1.10 Niveles de disponibilidad (Continuación)

1.5 PREPARANDO A LA ORGANIZACIÓN PARA LA ALTA DISPONIBILIDAD

Frecuentemente, uno de los grandes obstáculos para la Alta Disponibilidad computacional no son los fallos en el *hardware* o *software*, sino la falta de procedimientos. El aspecto humano de un Sistema de Alta Disponibilidad siempre será una fuente adicional de puntos de fallo. De esta forma, es esencial desarrollar una organización que vea la Alta Disponibilidad como la principal prioridad y que canalice los recursos necesarios que demande el ambiente de Alta Disponibilidad.

1.5.1 DEFINICIÓN DE METAS

Primeramente, habrá que definir las metas de manera explícita, un Acuerdo de Nivel de Servicio entre el analista-diseñador de Sistemas de Alta Disponibilidad y los usuarios del sistema será la manera de reunir los requerimientos de disponibilidad en términos de los servicios que son proporcionados a los clientes. El Acuerdo de Nivel de Servicio comprende los periodos normales de operación del sistema, la lista de las interrupciones planeadas y requerimientos específicos de rendimiento.

Por ejemplo, algunos elementos que pueden encontrarse en un Acuerdo de Nivel de Servicio están expuestos en la Tabla 1.11:

El sistema será 99.5% disponible durante 24 horas, 5 días de la semana y 52 días del año
El tiempo de respuesta a los clientes será de 1 a 2 segundos excepto durante periodos donde realicen respaldos incrementales
Los respaldos completos serán llevados a cabo cada fin de semana como mantenimiento planeado requiriendo 90 minutos para ello
Respaldos incrementales en línea tomarán lugar una vez cada día durante la semana de trabajo con un tiempo de respuesta de 2 a 3 segundos durante 30 minutos.
El tiempo de recuperación seguido de una falla no será mayor a 5 minutos

Tabla 1.11 Algunos elementos que deberán encontrarse en un Acuerdo de Nivel de Servicio

El Acuerdo de Nivel de Servicio es una especie de contrato entre el proveedor de soluciones de Alta Disponibilidad y el usuario, donde también se determinan de manera explícita los recursos necesarios para proveer de manera satisfactoria el servicio. Si es posible se identificará el costo-beneficio de la solución.

1.5.2 CONSTRUIR EL AMBIENTE FÍSICO APROPIADO

Diseñar un Sistema de Alta Disponibilidad requiere atender el ambiente físico. Es importante controlar el ambiente tanto como sea posible, con el fin de evitar problemas tales como el sobrecalentamiento y la rotura de cables. Además, los Sistemas de Alta Disponibilidad debieran ser físicamente seguros, bajo candado y llave, y disponibles solo a personal autorizado.

1.5.3 CREACIÓN DE PROCESOS AUTOMATIZADOS

Los errores provocados por el factor humano son siempre impredecibles, por ello, habrá que automatizar tantos procesos como sea posible; algunos de estos procesos pudieran ser los listados en la Tabla 1.12:

Respaldos rutinarios
Mantenimiento rutinario
Actualización del *software*
Recuperación ante fallo

Tabla 1.12 Procesos candidatos a ser automatizados

La automatización ayuda prevenir interrupciones y ayuda a restaurar los servicios tan rápido como sea posible cuando una interrupción ocurre. El monitoreo automático de los procesos que se ejecutan en el Sistema de Alta Disponibilidad puede detectar más rápido los problemas y notificar la acción correctiva necesaria. Un monitor automático además de detectar la falla, puede registrarla, solucionarla y anotar la acción llevada a cabo.

1.5.4 USO DE UN AMBIENTE DE DESARROLLO Y PRUEBAS

Antes de ejecutar nuevo *software*, éste deberá ser desarrollado y probado en ambientes creados para ello. Asimismo resulta necesario aislar el Sistema de Alta Disponibilidad contra estas prácticas para evitar que el sistema en producción se vea comprometido.

1.5.5 MANTENIMIENTO DE UNA CAJA DE REPUESTOS

Una práctica muy útil es tener siempre a la mano una caja de repuestos de partes que puedan ser empleadas cuando exista una falla de *hardware*. Esta caja deberá incluir discos duros, fuentes de poder, tarjetas y otros componentes de red que sean considerados parte indispensable de esta caja para subsanar lo más pronto posible cualquier emergencia que se presente.

1.5.6 DEFINICIÓN DE PROCEDIMIENTOS

Cuando un problema ocurre, los administradores del sistema (o sistemas) y los operadores deberán saber solucionarlo de acuerdo al curso de las acciones. Por ejemplo, la Tabla 1.13 muestra algunas de estas situaciones:

Cuando una recuperación automática se lleva a cabo
Cuando el sistema falla y requiere la intervención de un operador o administrador
Cuando existe una llamada de soporte
Cuando sucede una recuperación ante desastre

Tabla 1.13 Situaciones posibles que los administradores y operadores deberán atender según la definición de procedimientos

1.5.7 PLANEACIÓN ANTE DESASTRES

Un aspecto final es el desarrollo de una estrategia clara para el tratamiento de desastres naturales. Bajo la presión de una catástrofe, tener un programa o procedimiento local o remoto puede hacer la gran diferencia.

1.5.8 CAPACITACIÓN PARA EL GRUPO DE ADMINISTRADORES

Los administradores de sistemas deberán estar capacitados para pensar en términos de Alta Disponibilidad, los procedimientos usados en Sistemas de Alta Disponibilidad frecuentemente son distintos de los empleados en sistemas convencionales. Los administradores y operadores deberán ser capaces de reconocer y tomar acciones correctivas ante un evento que atente contra la Alta Disponibilidad.

1.5.9 USAR ESCENARIOS SIMULADOS

Una manera de capacitar al personal es la creación de escenarios, donde se simule un problema y los pasos a través de los cuales se llegaría a su solución.

1.5.10 DOCUMENTACIÓN DETALLADA

Otro elemento importante es la documentación de cada detalle en cuanto a la configuración del Sistema de Alta Disponibilidad y la creación de un manual de procedimientos que sea periódicamente revisado y actualizado por los administradores del sistema. Otro documento importante que deberá ser frecuentemente revisado es la bitácora de toda la actividad excepcional del Sistema de Alta Disponibilidad. Esta bitácora deberá contener las entradas generadas por un problema y registradas en el archivo que genera el sistema de Alta Disponibilidad así como las acciones correctivas llevadas a cabo cuando esta entrada apareció, de igual forma tanto las interrupciones planeadas como las no planeadas deberán ser registradas.

1.6 MIGRACIÓN A UN SISTEMA DE ALTA DISPONIBILIDAD

Los sistemas convencionales pueden migrar a Sistemas de Alta Disponibilidad a través de maneras un tanto severas, la Tabla 1.14 muestra estos caminos:

Por redundancia de componentes
Empleando *software* y *hardware* que emplee técnicas que permitan conmutarse de tal forma que siempre esté disponible su servicio
Planificando de manera cuidadosa todas las interrupciones planeadas
Eliminando la interacción del ser humano con el sistema
Definiendo las respuestas automáticas ante errores y eventos
Definiendo y practicando la respuesta de los administradores y operadores ante interrupciones no planeadas cuya respuesta no pueda ser automatizada

Tabla 1.14 Caminos para migrar hacia los sistemas de Alta Disponibilidad

De cualquier manera es imprescindible tener presente que cualquier diseño de un Sistema de Alta Disponibilidad deberá ser fuertemente probado antes de ponerlo en producción.

1.7 CREACIÓN DE UN SISTEMA DE ALTA DISPONIBILIDAD

Actualmente en el mercado de la Alta Disponibilidad existen diversas soluciones, las cuales provienen tanto de la iniciativa privada como de la comunidad del *Software Libre*. Como es de esperarse algunas de estas soluciones no cubrirán todos los requerimientos de ciertos clientes por no ser soluciones a la medida.

En esta parte del capítulo mostraré cómo podemos crear un Sistema de Alta Disponibilidad, basándonos en un método extremadamente efectivo como es el uso de *clusters*. El presente estudio se centra en la técnica de obtener una Alta Disponibilidad por medio de la redundancia, instalando dos o más servidores completos en lugar de uno solo, que sean capaces de trabajar en paralelo y de asumir las caídas de algunos de sus compañeros, y podremos añadir y quitar servidores al grupo (*cluster*) según las necesidades. A esta técnica se la denomina *clustering*.

1.7.1 INTRODUCCIÓN AL CLUSTERING DE SERVIDORES

En 1994 *Thomas Sterling* y *Don Becker*, trabajando para el CESDIS (*Center of Excellence in Space Data and Information Sciencies*) bajo el patrocinio del Proyecto de las Ciencias de la Tierra y el Espacio (ESS) de la NASA, construyeron un cluster de computadoras que consistía en 16 procesadores DX4 conectados por una red *Ethernet* a 10Mbps. Ellos llamaron a su máquina *Beowulf*. La máquina fue un éxito de inmediato y su idea de proporcionar sistemas basados en equipos de escritorio para satisfacer requisitos específicos de cómputo se propagó rápidamente a través de la NASA y en las comunidades académicas y de investigación. El esfuerzo del desarrollo para esta primera máquina creció rápidamente en lo que ahora llamamos el Proyecto *Beowulf*.

Beowulf, construido en la NASA en 1994, fue el primer *cluster* de la historia, y su finalidad era el cálculo masivo de datos. Desde entonces, la tecnología de *clusters* se ha desarrollado enormemente, apareciendo gran cantidad de estudios, teorías, programas y arquitecturas implantando *clusters* para diversos fines.

En general, podríamos decir que hay dos tipos de *clusters*, atendiendo a su finalidad:

Tipo 1 - Clusters para el procesamiento masivo de datos: El ejemplo más claro de este tipo es el Proyecto *Beowulf*. Este tipo de *clusters*, por lo general aprovechan la posibilidad de paralelización de cierto tipo de operaciones matemáticas (en especial, el cálculo matricial) para repartir los datos entre todos los equipos del *cluster* y poder así operar varios grados de magnitud más rápido. Para este fin se utilizan librerías como las PVM (*Parallel Virtual Machine*), que facilitan la distribución de datos entre las máquinas, incluso entre máquinas con distintos sistemas operativos, arquitecturas y lenguajes de programación.

Otro ejemplo de *cluster* de este tipo sería el caso de MOSIX, unos parches para el núcleo de Linux con los que se consigue poder utilizar de forma transparente toda una red de equipos como si fuera una única supercomputadora, permitiendo el migrado transparente de cara al usuario de procesos de una máquina a otra y la compartición de recursos.

Toda la teoría sobre este tipo de *clusters* se centra en cómo compartir los recursos de procesador, memoria y/o red entre los equipos que forman el *cluster* para obtener un mejor rendimiento general.

Tipo 2 - Clusters de Alta Disponibilidad: En este caso lo que se busca no es exactamente conseguir una gran potencia de cálculo sino conseguir un conjunto de máquinas que todas realicen la misma función y que, ante la falla de una de ellas, las demás puedan asumir sus tareas de una forma transparente y rápida. Por supuesto, la escalabilidad también es importante, ya que siempre podremos añadir más máquinas al *cluster* para así conseguir más potencia, pero el objetivo prioritario no es éste, sino la resistencia a cualquier fallo imprevisto.

Aquí lo que se busca con la replicación de máquinas es evitar los Puntos Simples de Fallo, que serían aquellas máquinas imprescindibles para el correcto funcionamiento del servicio que queremos dar; si únicamente tenemos una de cada máquina de este tipo, se convierte en un Punto Simple de Fallo y ante cualquier fallo en este equipo, todo el *cluster* queda inutilizado. La teoría sobre este tipo de *clusters* gira en torno a estos Puntos Simples de Fallo y cómo evitarlos, mediante la redundancia de *hardware* y el *software* apropiado para controlar el correcto funcionamiento de todos los equipos y, en caso negativo, hacer que una máquina de respaldo suplante a la que acaba de fallar.

1.7.2 ARQUITECTURA BÁSICA DE CLUSTERS DE ALTA DISPONIBILIDAD

Existen tres arquitecturas básicas de *clusters* de Alta Disponibilidad (ver Tabla 1.15):

ARQUITECTURA	DESCRIPCIÓN
Activa-Pasiva(Ver Figura 1.7)	Una arquitectura Activa-Pasiva es una en la cual un SPU pasivo es configurado para que tome actividad después del fallo del otro SPU. Dos o más SPU's son conectados a los mismos discos de datos, si el SPU falla, la aplicación debe iniciar en el SPU pasivo. El fallo en el SPU activo puede entonces ser atendido. En el SPU pasivo puede ejecutarse algún proceso o simplemente esperar el fallo del otro SPU.
Activa-Activa(Ver Figura 1.8)	Una arquitectura Activa-Activa es una en la cual los nodos pueden ejecutar aplicaciones de misión crítica, y alguno puede servir como respaldo de otros mientras ejecuta sus propias aplicaciones.
Base de datos paralela	Es un *cluster* en el cual los diferentes nodos ejecutan por separado, instancias de la misma base de datos y todos los accesos son concurrentes. En esta arquitectura la pérdida de un nodo no es crítica.

Tabla 1.15 Arquitecturas básicas de un cluster de Alta Disponibilidad

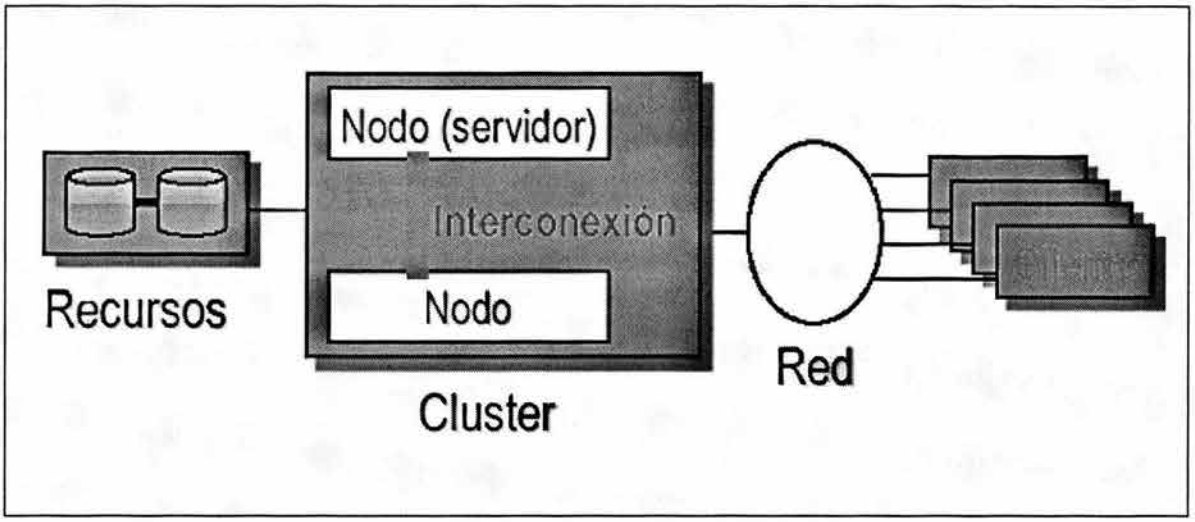

Figura 1.7 Arquitectura de cluster Activa-Pasiva

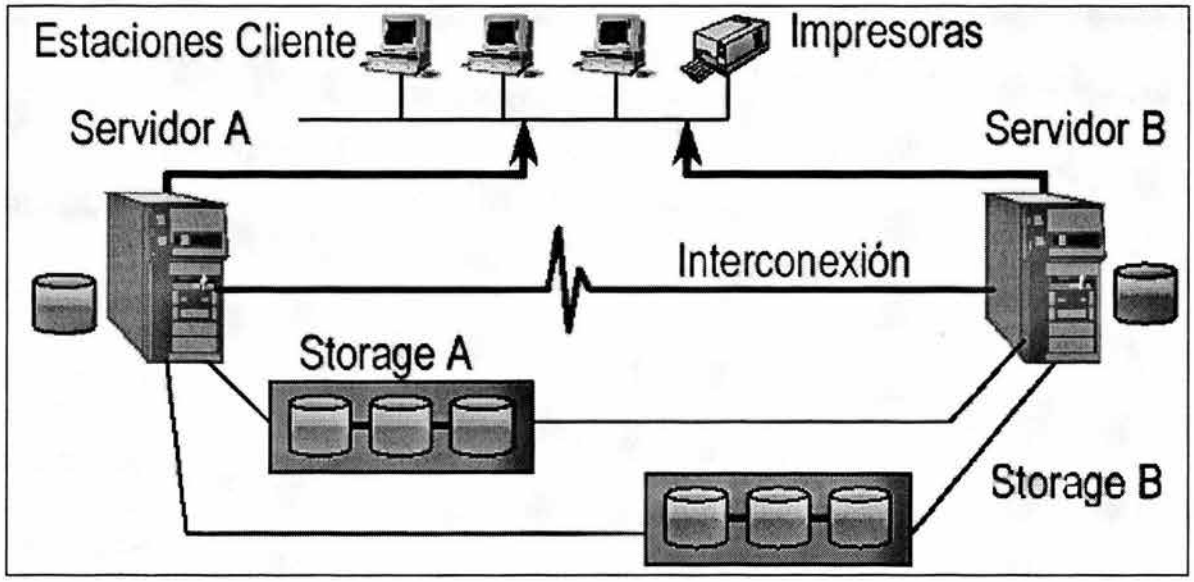

Figura 1.8 Arquitectura de cluster Activa-Activa

1.7.3 DINÁMICA DE LOS SISTEMAS DE ALTA DISPONIBILIDAD

Se considera dinámica de los Sistemas de Alta Disponibilidad a todas las configuraciones del *cluster* que garanticen la máxima disponibilidad del grupo de servicios. Esta dinámica está orientada a los nodos integrantes del *cluster* y la forma en la cual el *cluster* responde.

Failover: Es un término genérico que se usa cuando un nodo debe asumir la responsabilidad de otro, importar sus recursos y levantar el Grupo de Servicios. Se ha de entender que una situación de *failover* es una situación excepcional, para la cual la Alta Disponibilidad ha sido concebida (el fallo de un nodo). Si sólo queda un nodo en el *cluster*, tras los fallos de los demás, estaremos en un Punto Simple de Fallo hasta que el administrador del sistema, verifique y restaure el *cluster*. También se ha de entender que el Grupo de Servicios sigue levantado, que es el objetivo de la Alta Disponibilidad.

Takeover: Es un *failover* automático, se produce cuando un nodo nota un fallo en el Grupo de Servicios. Para ello debe haber cierta monitorización con respecto al Grupo de Servicios. El nodo que se declara averiado es forzado a ceder el servicio y recursos, o simplemente es eliminado.

Switchover o Giveaway: Es un *failover* manual, consiste en ceder los recursos de un Grupo de Servicios a otro nodo del *cluster*, mientras se realizan ciertas tareas administrativas. A este procedimiento se le denomina *node outage*.

Splitbrain: Para la gestión de un *cluster* de Alta Disponibilidad es necesario un mecanismo de comunicación y verificación entre nodos integrantes. Por este mecanismo, cada nodo debe gestionar los recursos que corresponden a cada uno, según el estado del cluster; a su vez cada nodo debe hacer verificaciones o latidos (*beats*) a sus compañeros. Un *Splitbrain* (división de cerebros) es un caso especial de *failover*, en el cual falla el mecanismo de comunicación y gestión de un *cluster* de dos nodos. Es una situación en la cual cada nodo cree que es el único activo, y como no puede saber el estado de su nodo compañero, tomará acciones en consecuencia, forzando un *takeover*. Esta situación es peligrosa, los dos nodos intentaran apropiarse de todos los recursos, incluyendo el Grupo de Servicios. El peligro aumenta sobre todo cuando tenemos recursos delicados, como recursos de almacenamiento, ya que cada nodo podría tomar y escribir por su cuenta, y quebrar la integridad de datos. Para evitar este problema, cada nodo debe actuar de una forma prudente, y utilizar los recursos compartidos como señal de que se está vivo. La forma de proceder de cada nodo es similar, ya que cada uno cree que su compañero ha desaparecido. Después de que un nodo aprecia este problema, tiene indicado que reserve un recurso llamado *quorum*. Un recurso *quorum*, es un recurso compartido, que se ha preestablecido en ambos nodos como tal. Éste es un recurso exclusivo, sólo un nodo del cluster puede reservarlo. Como este recurso sólo puede ser reservado por un nodo, el nodo que llegue

tarde a la reserva del recurso, entiende que debe abandonar el *cluster* y ceder todos sus recursos. El *quorum* es utilizado simplemente como un método de decisión. Otra forma de evitar esta situación, aunque un poco mas violenta, es que un nodo elimine a su compañero; el primero que apague a su compañero se queda con todos los recursos. Es un mecanismo muy brusco, pero muy eficaz. Para este caso existen tarjetas especializadas en esta tarea.

1.7.4 GRUPO DE SERVICIOS Y GRUPO DE RECURSOS

El Grupo de Servicios y sus interrupciones es lo que nos ha llevado a adoptar medidas con Tolerancia a Fallos o Alta Disponibilidad. En Alta Disponibilidad, se denomina Grupo de Servicios a un determinado servicio y los recursos necesarios para que el servicio sea ofrecido a un cliente (Grupo de Recursos). En otros entornos de Alta Disponibilidad, se denominan *logical host* o *software packages* a la unión de Grupo de Servicios y Grupo de Recursos asociados a éste.

Estos Grupos de Recursos han de implementar los mecanismos necesarios, para que sean completamente flexibles entre nodos del *cluster*, es decir puedan ser suplantados o conmutados físicamente entre los restantes nodos sin que el Grupo de Servicios cambie en absoluto. Esta virtualización del recurso va a permitir que un nodo pueda suplantar la función de otro dentro del *cluster*. El Grupo de Servicios sólo será penalizado con el tiempo de conmutación necesario para los Grupos de Recursos y la puesta en marcha del propio Grupo de Servicios.

Pensando en un Grupo de Servicios cualquiera, se puede pensar en una serie de recursos que va a utilizar. Por ejemplo, imaginemos un Grupo de Servicios del servicio Web, éste va a necesitar un nodo donde va a ser ejecutado (ciclos de CPU, memoria), un sistema de archivos donde guardar toda la información Web y por supuesto una red donde pueda responder a las peticiones de servicio. Estos recursos deben atender a cierta flexibilidad, donde la flexibilidad es la capacidad del recurso de ser estático virtualmente y ser dinámico físicamente.

Recursos computacionales: Los recursos computacionales pueden ser considerados a nivel de SPU, nodo o *cluster*. Son los recursos que permiten que el programa que se encarga de ofrecer el Grupo de Servicios pueda ser ejecutado. Si tenemos varios SPU's, varios nodos o varios *clusters* éstos deberán tener una copia del programa del Grupo de Servicios en memoria.

En Alta Disponibilidad, este recurso se considera a nivel de nodo, donde el Grupo de Servicios va a estar situado en un nodo determinado. El software de Alta Disponibilidad es quien decide que nodo va a alojar que Grupo de Servicios dependiendo del estado del *cluster*.

Recursos de comunicaciones: Normalmente el Grupo de Servicios va a ser accedido mediante una red de comunicaciones. Las tarjetas de red así como la pila de protocolos de red, deben ser capaces de responder a varias direcciones de red con el fin de dar flexibilidad al Grupo de Servicios; es decir virtualizar el servicio. En el caso de redes TCP/IP el Grupo de Servicios será accedido mediante una dirección IP y un puerto; para que el grupo de servicios pueda residir físicamente en cualquier nodo se deben utilizar IP's virtuales.

Recursos de almacenamiento: El almacenamiento de los datos del Grupo de Servicios es quizás uno de los puntos más delicados de la Alta Disponibilidad. El almacenamiento suele ser el recurso mas complicado de virtualizar en configuraciones clásicas; ya que suele ser un medio SCSI compartido con muchos discos y muchos elementos candidatos a Puntos Simples de Fallo. En configuraciones modernas de *hardware* no hay tanta problemática; las arquitecturas SAN (*Storage Area Network*) permiten que los recursos de computación accedan por red SAN a los recursos de almacenamiento. Los recursos de almacenamiento suelen ser servidores de archivos con discos en RAID y respaldo integrado con interfaz *FiberChannel.* Al estar en un entorno SAN permite acceder a estos recursos a más de un nodo de forma muy flexible. El recurso de almacenamiento además de ser flexible debe ser independiente para las necesidades de cada Grupo de Servicios. Aquí surge el concepto de Grupo de Volúmenes, el cual es un conjunto de volúmenes que pertenecen a un Grupo de Servicios en concreto; cada volumen es un espacio de almacenamiento que el Grupo de Servicios utilizará para sus propósitos con independencia de otros volúmenes. Es vital que el recurso de almacenamiento sea capaz de mantener la integridad de los datos y el tiempo de recuperación ante un fallo sea mínimo.

Journaling es una técnica que nació de las bases de datos y como respuesta a los problemas comentados previamente para dar solución a la gestión de datos, técnica que se ha ido incorporando a los sistemas de archivos con muy buenos resultados. Cuando un sistema de archivos sufre una caída, dado a un fallo del sistema, éste se verifica a sí mismo por completo corrigiendo las inconsistencias presentes. Con *journaling* se lleva una cuenta de qué se ha ido modificando en el sistema de archivos, de manera que a la hora de verificarlo sólo comprobará las inconsistencias de unos pocos archivos y directorios.

1.7.5 PUNTOS SIMPLES DE FALLO

En cualquier Sistema de Alta Disponibilidad, es importante entender los diferentes tipos de posibles fallos que se pueden presentar y cómo el sistema puede darles respuesta. No todas las interrupciones son causadas por fallos, pero todos los fallos definitivamente causarán interrupciones a menos que podamos interceptarlos.

La disponibilidad puede verse como una cadena de servicios que debiera permanecer unida. Los fallos son roturas en la cadena. Los eslabones débiles son conocidos como Puntos Simples de Fallo (*Single Points Of Failure*, SPOF). Por cada eslabón en la cadena que es un posible punto de fallo, podemos reducir la oportunidad de interrupción proporcionando un respaldo o eslabón alternativo. Este proceso es llamado Eliminación de Puntos Simples de Fallo (*Eliminating Single Points Of Failure*) en el sistema.

Es decir, la regla básica para hacer un Sistema de Alta Disponibilidad es la replicación de elementos. Con Puntos Simples de Fallo se quiere hacer referencia a cualquier elemento no replicado y que puede estar sujeto a fallos; afectando con ello al servicio. Se debe evitar que existan Puntos Simples de Fallo en cualquier subsistema del sistema, ya que en caso de fallo puede peligrar el Grupo de Servicios por culpa de un subsistema.

Por ejemplo, un adaptador de red que da acceso a un servidor a una red es un Punto Simple de Fallo, una controladora SCSI también lo es. Si en un entorno de servidores falla uno y no puede ser reemplazado fácilmente por otro, el servidor se considerará como un Punto Simple de Fallo.

En el caso de Alta Disponibilidad (con *hardware* y *software* adecuado) se puede reemplazar un adaptador de red o un servidor automáticamente. No sólo los servidores han de ser redundantes sino también los elementos que facilitan el servicio, como los *routers*, *bridges* y hasta la propia red local.

Es conveniente olvidar la redundancia a cierto nivel, pues habrá un nivel en que la Alta Disponibilidad se hace extremadamente cara, tal es el caso que podría considerarse entonces que si sólo tenemos un único edificio para nuestro sistema, perfectamente podemos considerarlo como Punto Simple de Fallo (en caso de incendio, sismo o cualquier otra amenaza que atente contra la seguridad del edificio), pero replicar en otro edificio el sistema, depende de la valoración del sistema mismo, del riesgo al que esté expuesto y hasta del mismo presupuesto.

1.7.6 IDENTIFICACIÓN DE PUNTOS SIMPLES DE FALLO EN UN SISTEMA AUTÓNOMO

Una vez aclarados ciertos conceptos, retomamos el objetivo del Tema 1.7, la primera fase de creación de un *cluster* de Alta Disponibilidad es la identificación y eliminación de los Puntos Simples de Fallo.

Un sistema autónomo altamente confiable todavía tiene muchos Puntos Simples de Fallo. Consideremos una arquitectura de red típica cliente-servidor (ver Figura 1.9).

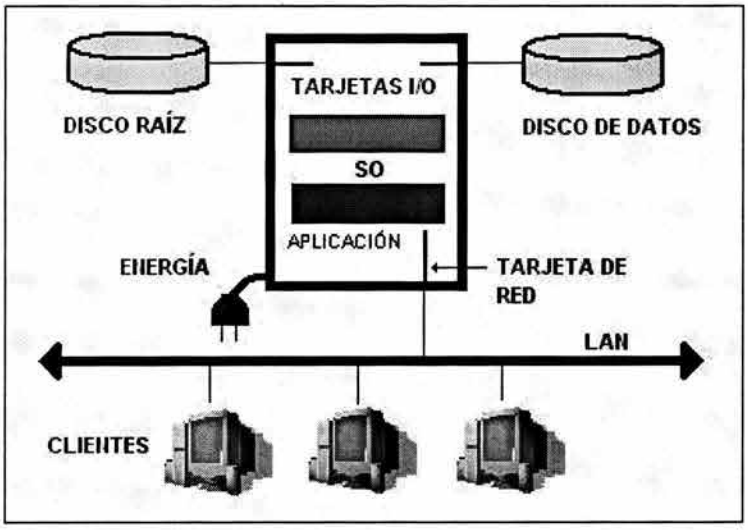

Figura 1.9 Sistema autónomo altamente confiable

Los clientes están ejecutando aplicaciones en computadoras personales o estaciones de trabajo. Además están conectadas a través de la red con el servidor de aplicaciones que está ejecutándose en un *System Processor Unit* (SPU). El servidor recibe y almacena la información proveniente de los clientes, esta información es almacenada en el disco de datos. El sistema operativo del servidor alojado en el disco raíz, controla las conexiones de sus clientes y la transferencia de datos, entre otras actividades.

¿Qué de malo puede acontecer en este escenario?

En la Tabla 1.16 tenemos algunos ejemplos de lo que podría acontecer:

El sistema causa baja por una falla en las tarjetas de expansión
El cable de red es dañado o la tarjeta de red falla y los clientes pierden el servicio
Seguido de un reinicio del sistema, un operador reinicia de forma incorrecta la aplicación y los clientes no pueden conectarse
Una falla en el disco raíz causa que el sistema pare
Una falla en el disco de datos causa la pérdida e interrupción del servicio
Una falla de la fuente de poder resulta en el reinicio del sistema y la pérdida de datos

Tabla 1.16 Posibles acontecimientos en un sistema autónomo altamente confiable

En la Tabla 1.17 se muestran las fallas típicas de estos componentes junto con una descripción de cómo el Punto Simple de Fallo puede ser eliminado.

COMPONENTE	¿QUÉ PASA SI EL COMPONENTE FALLA?	¿CÓMO ELIMINAR EL PUNTO SIMPLE DE FALLO?
SPU simple	El servicio está perdido hasta que el SPU esté reparado	Proveer un SPU de respaldo con la creación de un *cluster*
Red de Área Local simple	La conectividad de los clientes se pierde	Instalar tarjetas de red y subredes redundantes
Tarjeta de red simple	La conectividad de los clientes se pierde	Instalar tarjetas de red redundantes o configurar tarjetas de red que estén a la espera de fallo
Disco raíz simple	El servicio es perdido hasta que el disco sea reemplazado	Usar un disco raíz redundante o *disk mirroring*
Disco de datos simple	La pérdida de datos	Usar almacenamiento redundante o *disk mirroring* o arreglos de discos
Fuente de poder simple	El servicio es perdido hasta que la energía sea restaurada	Usar fuentes de poder adicionales y emplear tecnología UPS
Interfaz SCSI simple	El servicio es perdido hasta que la interfaz sea reemplazada	Redundancia
Sistema Operativo	El servicio es perdido hasta que el sistema operativo reinicie	Proveer capacidades ante caída del sistema y para el reinicio y recuperación de las aplicaciones
Aplicaciones	El servicio es perdido hasta que las aplicaciones reinicien	Proveer la facilidad para reiniciar la aplicación
Ser Humano	El servicio es perdido hasta que el error humano sea corregido	Automatizar las operaciones lo más posible, Documentar cada uno de los procedimientos

Tabla 1.17 Eliminando Puntos Simples de Fallo

1.7.7 ELIMINACIÓN DE FUENTES DE PODER COMO PUNTOS SIMPLES DE FALLO

La Figura 1.8 muestra todos los componentes de un sistema conectado a una sola fuente de poder, obviamente es un Punto Simple de Fallo el cual puede ser corregido de muchas formas. Un *Uninterruptible Power Suply* (UPS) o como lo conocemos dentro del argot computacional como *No-Break* proporciona una fuente de poder en espera de una interrupción en la fuente de poder original. Pequeñas unidades UPS pueden ser usadas para proteger SPU's individuales. Esto permite que los procesos continúen hasta que la energía regrese o hasta que suceda un apagado normal del sistema. Un pequeño UPS puede proteger a pocos componentes del sistema, pero el UPS por si solo puede fallar también. Por esta razón, diversas UPS deberán ser usadas para proteger diferentes partes del sistema. La Figura 1.10 muestra los dispositivos conectados a dos diferentes circuitos.

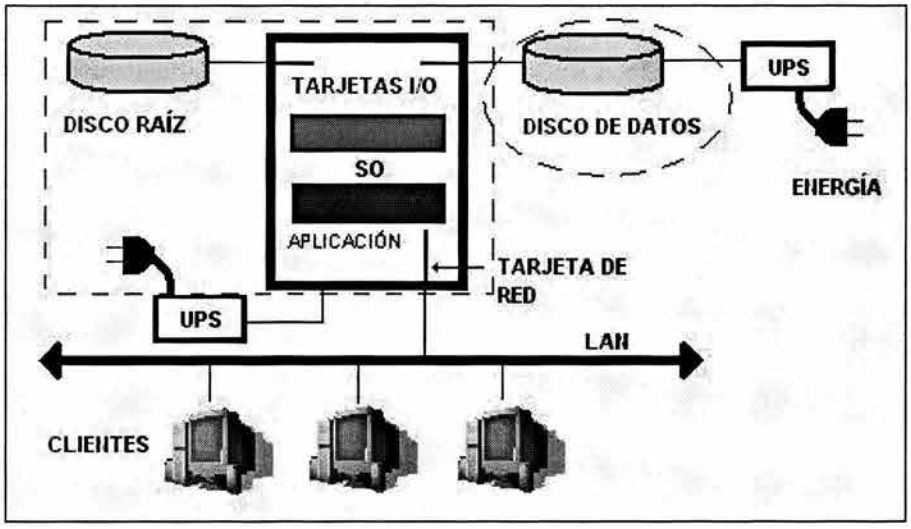

Figura 1.10 Empleo de más de una unidad UPS

Las líneas punteadas muestran los dispositivos que son protegidos por cada UPS. Instalaciones pequeñas pueden encontrar el uso adecuado de UPS's individuales para proteger la integridad del sistema durante una falla corta de energía. En particular, la UPS debería proveer la energía suficiente para que el sistema pueda completar todas las transacciones, sincronizar todas las bitácoras y permitir un apagado normal del sistema.

1.7.8 ELIMINACIÓN DE DISCOS COMO PUNTO SIMPLE DE FALLO

Otro Punto Simple de Fallo que resulta obvio corresponde a los discos en un sistema autónomo. En la Figura 1.9 se pueden apreciar dos discos: raíz y de datos. Si existe una falla en el disco raíz, el sistema está imposibilitado para continuar con los procesos. El disco deberá ser reemplazado al igual que su contenido reinstalando las aplicaciones y/o a través de un respaldo. Si el disco de datos falla, el sistema puede restaurarse, pero las aplicaciones pararán hasta que un nuevo disco pueda ser instalado y los datos sean recuperados. Ante la falla de un disco, el sistema deberá ser reiniciado y los datos recuperados de los respaldos. La redundancia es necesaria para prevenir que el fallo de un disco provoque una interrupción del servicio.

Una técnica para proveer redundancia es el uso de arreglos de discos con configuraciones RAID (*Redundant Array of Inexpensive Disks*). Un grupo de discos funciona con una variedad de arreglos conocidos como niveles RAID. Algunos niveles permiten la réplica de *hardware*, mientras que otros proveen protección a través del uso de paridad de datos, el cual permite al arreglo la reconstrucción de los datos perdidos si un disco falla.

Los niveles RAID más comunes son los que se muestran en la Tabla 1.18:

NIVEL	DESCRIPCIÓN
0	Permite que múltiples discos sean vistos como una única unidad lógica mediante una técnica denominada *drive spanning*, de forma que la capacidad de la unidad lógica es igual a la suma de las capacidades de todas las unidades físicas. Se puede usar con cualquier número de discos físicos (de dos en adelante) limitados sólo por la capacidad de nuestra controladora. Para distribuir los datos entre los diferentes discos físicos se usa otra técnica denominada *drive striping* que maximiza el rendimiento de las operaciones de entrada-salida. Para ello, se divide el disco lógico en bloques de datos denominados bandas (*stripes*), las cuales se distribuyen entre los discos físicos. Durante las operaciones de lectura y escritura los discos operan simultáneamente.
1	Emplea la técnica denominada *drive mirroring*, mediante la cuál creamos un único disco lógico usando para ello dos (y sólo dos) discos físicos. Todos los datos que escribimos en el disco lógico son escritos en ambos discos físicos, de forma que ambos son, en todo momento, gemelos. El espacio real disponible se reduce, pues, al 50%. El rendimiento en la lectura de datos se incrementa, pero empeora en la escritura. RAID 1 nos proporciona un buen nivel de tolerancia a fallos y de rendimiento, pero la peor eficiencia en cuanto al espacio de almacenamiento disponible. Cuando usamos RAID 1 con dos controladoras de disco independientes, la técnica resultante se denomina *drive duplexing* y nos proporciona uno de los máximos niveles de tolerancia a fallos que podemos lograr en este aspecto.

Tabla 1.18 Niveles RAID

NIVEL	DESCRIPCIÓN
1E o 6	El RAID 1E (*enhaced*) combina las técnicas de *mirroring* y *striping* de forma que nuestro disco lógico es igualmente dividido en bandas, de forma que cada una de ellas está escrita en dos discos distintos. De esta forma podemos permitir cualquier número de discos físicos y no sólo dos como en el RAID 1. El espacio útil sigue reducido al 50% de la capacidad total y todo lo dicho en cuanto al rendimiento de lecturas y escrituras del RAID 1 es válido también para este esquema.
10 o 1+0	Combina también, aunque de distinta forma, las técnicas de *mirroring* y *striping*. Es el resultado de realizar un *mirroring* de dos volúmenes. El número de discos usados ha de ser par, la capacidad de espacio útil es del 50% y tenemos rendimientos de lectura y escritura similares a los proporcionados por RAID 0.
3	RAID 3 requiere al menos tres discos físicos. Uno de ellos está dedicado exclusivamente a almacenar la paridad de los datos de todos los demás. Los datos se encuentran, al igual que en esquemas anteriores, divididos en bandas. Usando paridad en lugar de *mirroring* estamos reduciendo considerablemente el espacio necesario para la redundancia de datos. Proporciona un alto rendimiento en operaciones de lecturas de grandes bloques y, como contrapartida, ocasiona un cuello de botella en las operaciones de escritura. RAID 3 está recomendado exclusivamente en las aplicaciones que requieran uso intensivo de lectura de datos y escasas escrituras. Este esquema y el siguiente (RAID 4) prácticamente no se usan en la actualidad, habiendo sido desplazados por RAID 5.
4	Es similar a RAID 3 con la única diferencia de que utiliza bandas más grandes para mejorar algo el rendimiento en las operaciones de escritura.
5	Este esquema usa bandas para almacenar los datos y paridad para proporcionar tolerancia a fallos. La principal diferencia respecto a RAID 3 y RAID 4 es que no dedica un disco en exclusiva para la paridad, sino que almacena ésta en bandas intercaladas entre los datos de todos los discos. Requiere un mínimo de tres discos y su eficacia en cuanto a espacio de almacenamiento es idéntica a la proporcionada por los dos RAID's anteriores. La distribución de las bandas de paridad entre todos los discos elimina el cuello de botella existente en las escrituras.

Tabla 1.18 Niveles RAID (Continuación)

Como puede apreciarse en la descripción de todos ellos, es necesario introducir nuevas variables a la hora de elegir cuál es el esquema RAID que más se ajusta a nuestras necesidades. Ya no nos basta con saber la disponibilidad que necesitamos y el costo que podemos asumir, tenemos que estudiar también el uso que harán nuestras aplicaciones de los discos. Por ejemplo, si se considera necesario hacer un uso intensivo de los discos para escritura el mejor esquema es el RAID 1. Si fundamentalmente vamos a realizar lecturas de grandes bloques de forma secuencial, RAID 3 o RAID 4 son los esquemas más apropiados. Sin embargo, si lo que se requiere es hacer uso intensivo de lectura de bloques de tamaño variado y de forma fundamentalmente aleatoria, obviamente los mejores esquemas son RAID 5 o RAID 10. Ante la duda, estos dos últimos esquemas son los más flexibles y los que ofrecen un mejor comportamiento medio ante cualquier situación.

Otro punto a tener en cuenta es la elección de los discos. Salvo en RAID 1, el resto de los esquemas proporcionan mayor velocidad de acceso y escritura y menor pérdida de capacidad utilizando discos pequeños en mayor cantidad, frente al uso de menor número de discos de mayor tamaño.

Otro punto importante a estudiar es el comportamiento de los distintos esquemas de RAID ante un estado crítico, es decir, tras haber perdido uno de los discos físicos. En RAID 1, ya sea *mirroring o duplexing*, si perdemos un disco el otro tiene una copia íntegra de todos los datos. Perdemos el beneficio adicional de lectura conjunta de ambos discos, pero no apreciaremos una sensible degradación del sistema. En RAID 3 o RAID 4, si uno de los discos de datos falla, su contenido será reconstruido a partir de la información almacenada en el disco de paridad, con lo que apreciaremos una sensible degradación del rendimiento. Si el disco que falla es el de paridad, el rendimiento del sistema no sufrirá ningún cambio apreciable. En RAID 5, la pérdida de cualquiera de los discos ocasionará una sensible pérdida de rendimiento. En todos ellos, la tolerancia a fallos desaparece hasta que no hemos reemplazado el disco dañado y éste ha sido reconstruido.

Cuando hablamos de las mejoras aportadas por un esquema RAID soportado por *hardware*, debemos mencionar dos funcionalidades llamadas *hot-swap* y *hot-spare*. Ambas tienen que ver con la forma de actuar cuando hemos perdido un disco. Si nuestro esquema RAID no está soportado por *hardware* específico, para reemplazar uno de ellos debemos necesariamente de apagar el sistema, reemplazar el disco, volver a iniciar el sistema además de la reconstrucción.

Si nuestro *hardware* de RAID soporta *hot-swap* podemos reemplazar el disco *"en caliente"* esto es, sin necesidad de apagar el sistema. Una vez que hemos introducido el nuevo disco inmediatamente comienza la reconstrucción del mismo. Durante todo el proceso de reconstrucción, sea cual sea el esquema de RAID que usemos, el rendimiento del sistema se verá sensiblemente degradado.

Hot-spare va un paso más allá. Si nuestro *hardware* soporta está técnica, podemos tener un disco adicional de reserva en nuestra cabina de discos. Cuando el *hardware* de RAID detecta que ha perdido uno de los discos útiles lo reemplaza inmediatamente por el de reserva e inicia la reconstrucción, de forma que el tiempo que permanecemos sin tolerancia a fallos es mínimo (únicamente mientras dure la reconstrucción del disco de reserva) y no se requiere en ningún momento una intervención manual para restablecer las condiciones de fiabilidad iniciales.

Aún tenemos un nivel mas de disponibilidad: los llamados *arrays of RAID arrays*. Éstos nos permiten la pérdida de un disco manteniendo la tolerancia a fallos y sin apreciar visiblemente ninguna degradación en el rendimiento del sistema, y la pérdida de dos discos simultáneamente manteniendo la disponibilidad del sistema. Además, estos esquemas tienen una excelente respuesta tanto en escritura como en lectura. Los principales inconvenientes son el elevado costo de implantación de los mismos y que estamos hablando de sistemas propietarios de diversos fabricantes de *hardware* sin ningún tipo de normalización al respecto.

Por último, en un nivel más alto y más cercano a la filosofía de respaldos, están soluciones como el SRDF (*Simmetrix Remote Data Facility*), distribuido por la casa *EMC2* y que consiste básicamente en una replicación a distancia de la totalidad o parte de nuestros dispositivos de almacenamiento. No precisa de CPU's dedicadas, es totalmente independiente de las máquinas, sistemas operativos y bases de datos que usemos en nuestra red, admite replicaciones síncronas o asíncronas configurables dinámicamente y casi cualquier medio de comunicación disponible en el mercado (*Frame Relay*, RDSI, ATM, etc).

En la Figura 1.11 se muestra un ejemplo de la eliminación de discos como Punto Simple de Fallo:

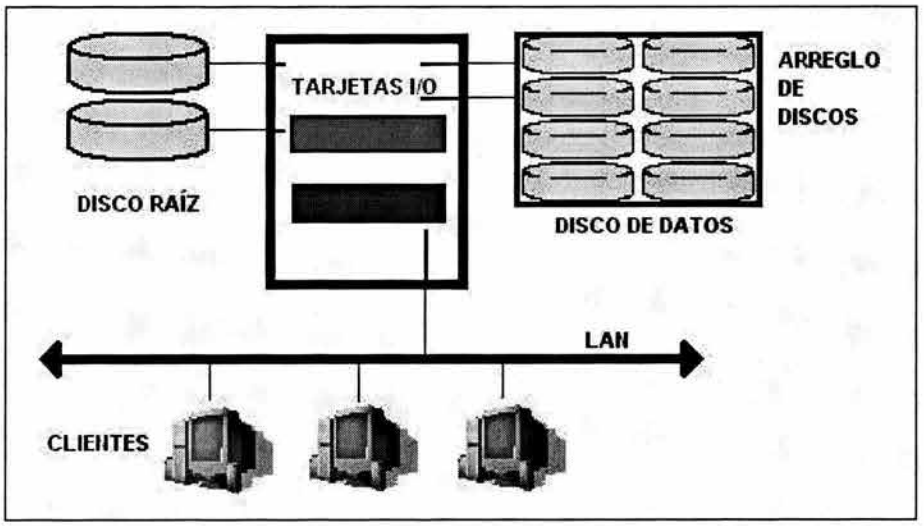

Figura 1.11 Ejemplo del uso de discos replicados (*disk mirroring*) y arreglo de discos

1.7.9 ELIMINACIÓN DEL SPU COMO PUNTO SIMPLE DE FALLO

El SPU consiste en un grupo de elementos, los cuales en cualquier momento pueden fallar. Los más importantes son él o los procesadores, los controladores de entrada-salida y los bloques de memoria, entre otros.

Si uno de estos componentes falla, el sistema típicamente tendrá que ser parado, los administradores tendrán que hacer un diagnóstico, tomar las medidas necesarias para su puesta en marcha e iniciar el sistema. De cualquier forma, la pérdida del servicio durante este intervalo puede ser inaceptable.

El uso de una arquitectura basada en *clustering* garantiza la eliminación del SPU como Punto Simple de Fallo. Un *cluster* elimina la interrupción asociada con la falla en el SPU, permitiendo a los administradores reparar o reemplazar los componentes defectuosos sin perder el servicio.

En un *cluster* de Alta Disponibilidad, uno o más sistemas actúan como respaldo y dichos sistemas pueden ser activos o pasivos. Los sistemas activos ejecutan sus propias aplicaciones mientras actúan como el respaldo de otro sistema. Los sistemas pasivos estarán inactivos hasta que una falla ocurra o sean usados por otros procesos.

La Figura 1.12 ilustra la adición de un segundo sistema, incluyendo el SPU, al sistema autónomo ilustrado en la Figura 1.8.

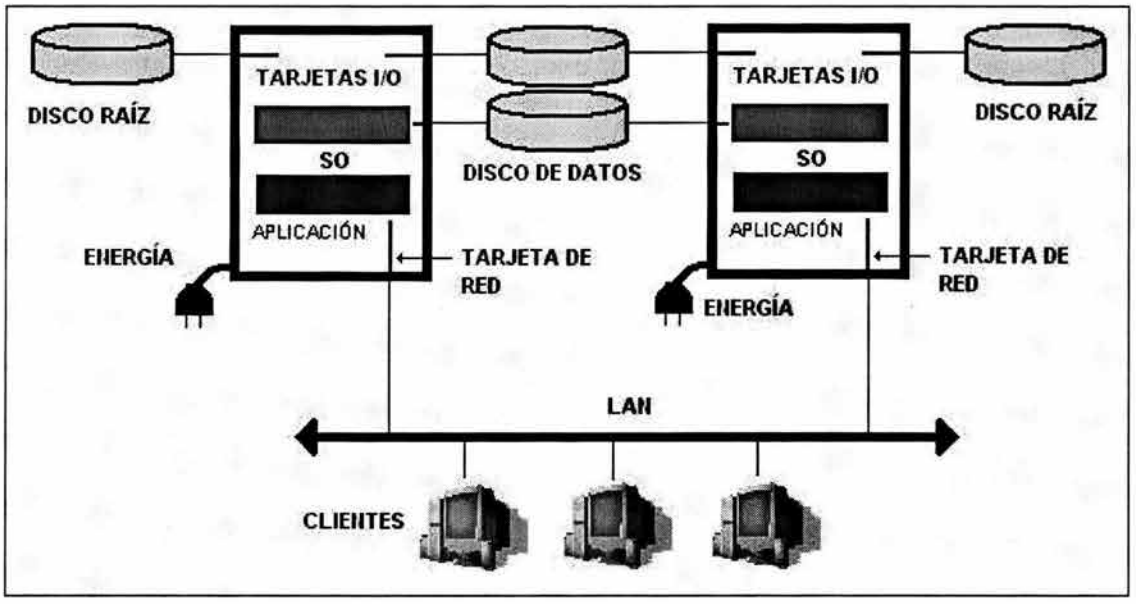

Figura 1.12 Cluster de dos nodos con SPU's redundantes

El resultado es un simple *cluster*, y los sistemas individuales son conocidos como nodos. Un nodo no es lo mismo que un SPU. El SPU contiene uno o más procesadores, memoria, y una fuente de poder. Un nodo es un sistema el cual es miembro de un *cluster*. EL SPU es un componente de un nodo.

Los dos nodos son conectados entre sí por una red de área local, la cual les permite aceptar las conexiones de los clientes y transmitir mensajes que confirman la "salud" de cada uno, es decir, que cada nodo hace una verificación de estado de su compañero. Si el SPU de un nodo falla, el otro nodo puede tomar su lugar después de un breve retardo, en un proceso conocido como *failover*. Después del *failover*, los clientes pueden tener acceso al segundo nodo de la misma forma que lo hacían con el primero.

El proceso de *failover* es manejado por una aplicación de Alta Disponibilidad ejecutándose en todos los nodos del *cluster*.

En la Figura 1.11 se puede apreciar que los discos de datos son compartidos por ambos nodos, esto es una opción, sin embargo existen otras opciones como la replicación de datos, es decir que cada nodo, tenga una réplica de los datos del original en todo momento; para esto existen actualmente diversas técnicas para lograr este objetivo, a nivel de software. Otra observación, cada nodo tiene su propio disco raíz.

1.7.10 ELIMINACIÓN DE LA RED COMO PUNTO SIMPLE DE FALLO

La red debiera usar componentes redundantes y en la medida de lo posible, el *hardware* de red debiera ser configurado con monitores que permitan detectar errores y reportarlos.

En *clusters* de Alta Disponibilidad, la red es configurada y usada para dos propósitos importantes, uno para el acceso a una aplicación por parte de los clientes u otros sistemas y el otro para la comunicación entre nodos del *cluster*.

La comunicación entre los clientes del sistema y el servidor está sujeta a varias clases de fallos. Dependiendo del tipo de *hardware* de red, los fallos pueden ocurrir en cables, tarjetas de red, *routers, hubs, switches* o concentradores.

La Figura 1.13 muestra una red LAN que cuenta con un "*Cluster* de Alta Disponibilidad":

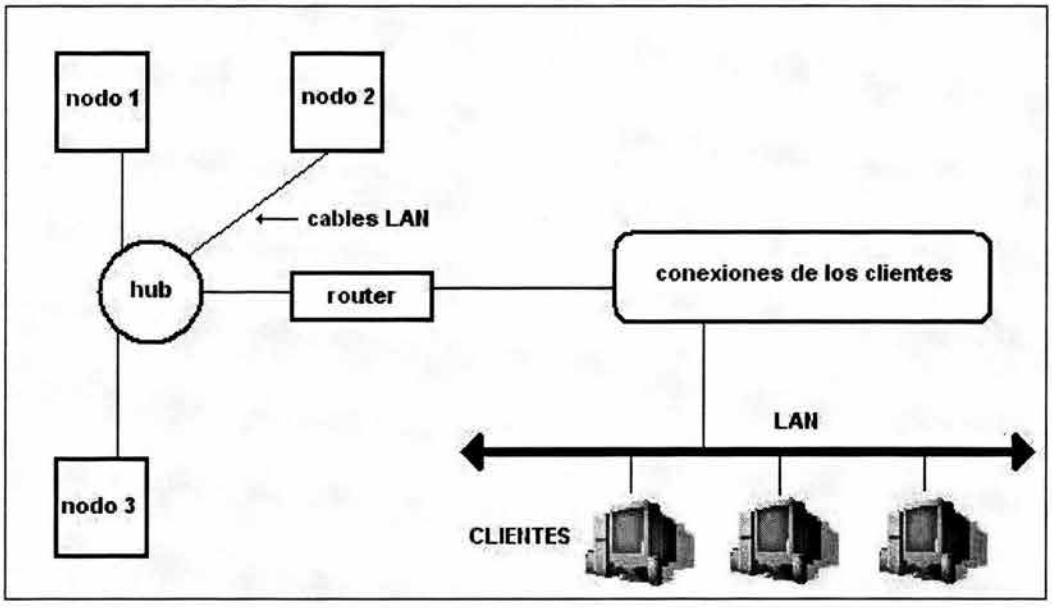

Figura 1.13 Configuración LAN de un "Cluster de Alta Disponibilidad"

La Figura 1.12 ilustra varios aspectos. La topología usada es de tipo estrella, tres nodos conectados a un *router*. Usando este tipo de configuración, los clientes pueden conectarse a los nodos, y los nodos pueden comunicarse entre sí también. Sin embargo, la configuración de la Figura 1.12 no es altamente disponible (ver Figura 1.14).

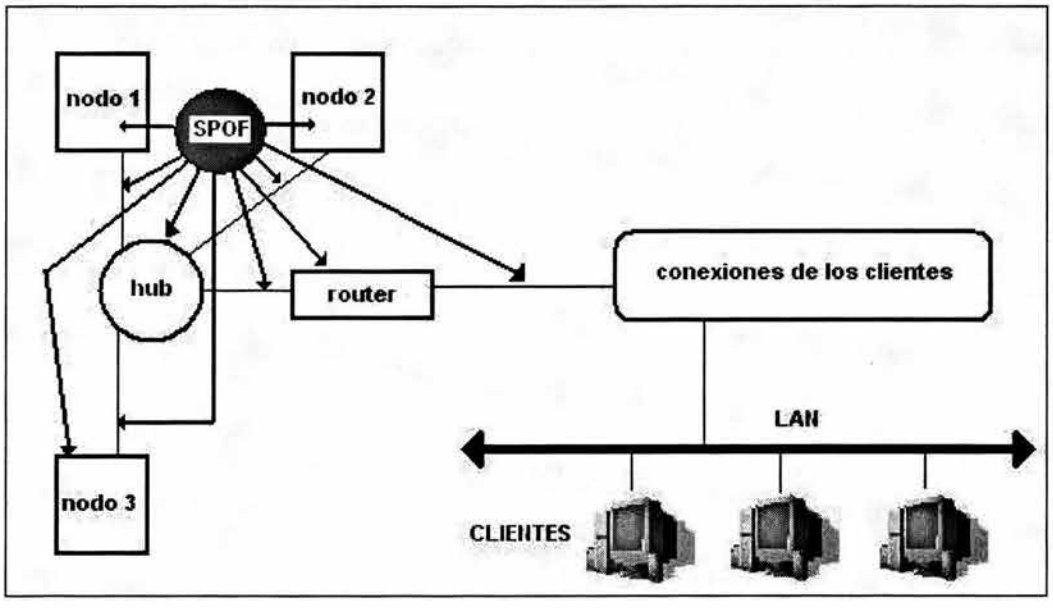

Figura 1.14 Puntos Simples de Fallo en la configuración LAN ilustrada en al Figura 1.12

Los Puntos Simples de Fallo son los ilustrados en la Tabla 1.19:

Clientes del sistema
Router
Hub Ethernet
Cables de red
Tarjetas de red

Tabla 1.19 Puntos Simples de Fallo de la configuración LAN mostrada en la Figura 1.12

El *cluster* fallaría si ocurre una falla en los cables y en las tarjetas de red. Sobre las conexiones de los clientes aún cuando pueden fallar, no constituye un Punto Simple de Fallo para el *cluster*, solo si ninguno de los clientes puede conectarse. En un *Cluster* de Alta Disponibilidad, el *software* de Alta Disponibilidad establece el vínculo de comunicación entre nodos, esto se conoce como *heartbeat*. Este mensaje o vínculo permite al *software* de Alta Disponibilidad decir si uno o más nodos han fallado. Este vínculo en la red debe ser especialmente protegido contra fallo. Podemos eliminar estos Puntos Simples de Fallo de red por medio de dos formas (ver Tabla 1.20):

Proporcionando conexiones LAN completamente redundantes
Configurando el intercambio local de tarjetas de red

Tabla 1.20 Formas de eliminar la red como Punto Simple de Fallo

La primera forma consiste en configurar de manera redundante el cableado estructurado y la redundancia en las tarjetas de red en cada nodo. Para eliminar la pérdida de conectividad de los clientes, podemos configurar *routers* redundantes. También podemos configurar un *hub* redundante para contrarrestar el fallo de otro *hub*. La Figura 1.15 muestra esta nueva configuración.

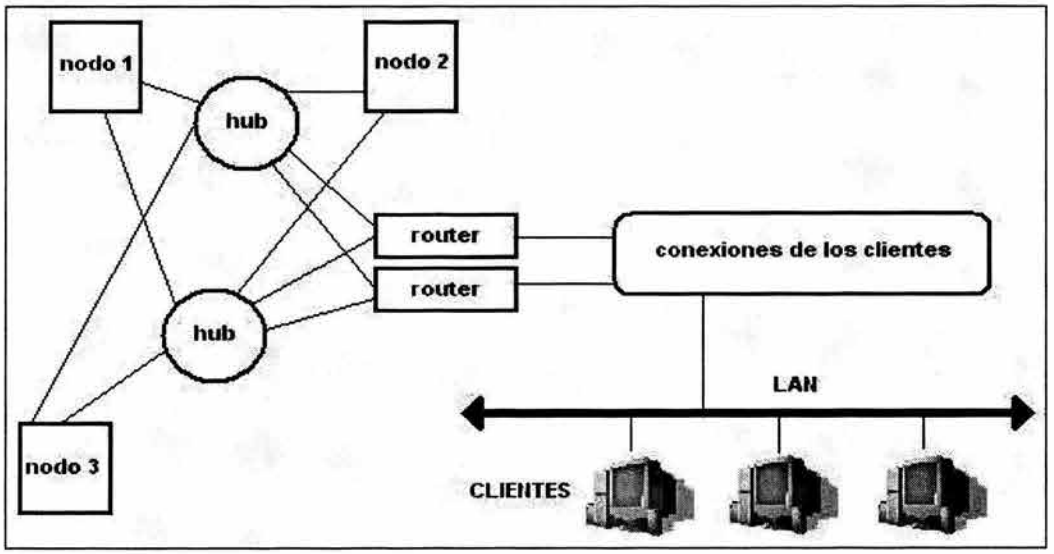

Figura 1.15 Alta Disponibilidad en la configuración LAN de la figura 1.12

Cabe aclarar que en algunas configuraciones, no podemos eliminar todos los Puntos Simples de Fallo.

Otra forma de eliminar Puntos Simples de Fallo es el intercambio local de tarjetas de red, que significa un cambio desde una configuración de tarjetas de red activas a otras pasivas cuando la red pierde conectividad. Esto es posible solo si se configuran las tarjetas de red pasivas por cada nodo, estas tarjetas deben estar en la misma subred que las tarjetas primarias. Esta subred es un dominio de conectividad física que consiste de segmentos LAN activos y pasivos conectados a través de un puente o *bridge*, de manera que ambas subredes estarán disponibles para soportar un evento en el cual tengan que intercambiarse, como lo ilustra la Figura 1.16:

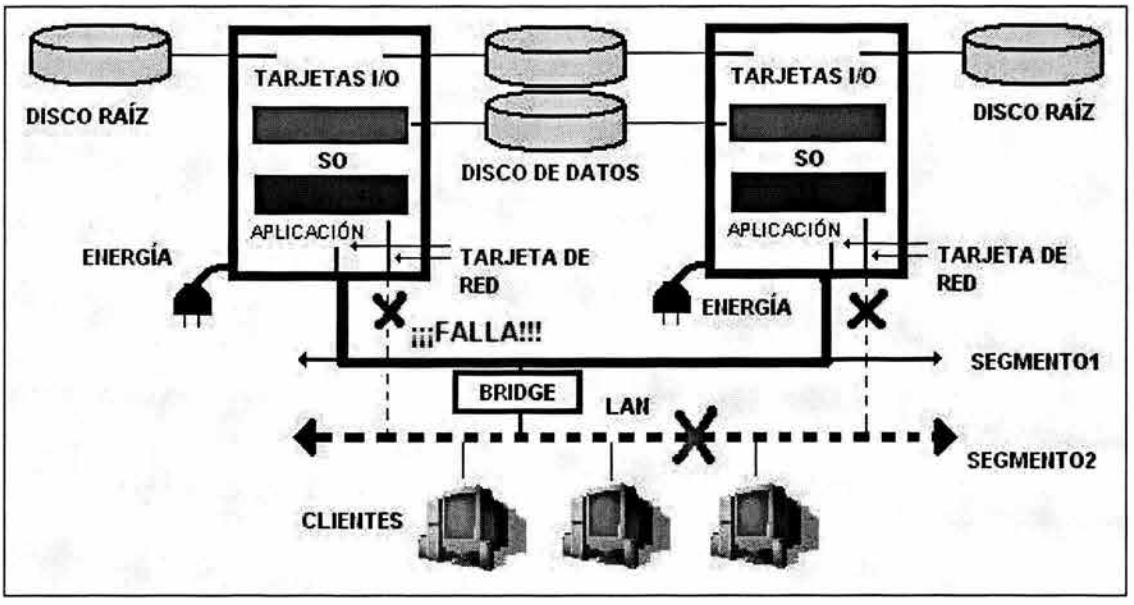

Figura 1.16 Ejemplo del intercambio de tarjetas de red como medio para eliminar la red como Punto Simple de Fallo

1.7.11 ELIMINACIÓN DE SOFTWARE COMO PUNTO SIMPLE DE FALLO

Muchas aplicaciones son susceptibles a fallar (ver Tabla 1.21).

Sistema operativo
Manejador de base de datos
Aplicaciones para el servidor
Aplicaciones para el cliente

Tabla 1.21 Aplicaciones susceptibles a fallos

Por ejemplo, si existe un fallo en el sistema operativo, el nodo causa baja y el *cluster* lo reconfigura de tal forma que los servicios que estaban disponibles en un nodo ahora lo están en otro. Una manera de hacer esto es tener otro nodo que tome las aplicaciones que estaban ejecutándose. En Alta Disponibilidad, la aplicación o aplicaciones son conocidas como Grupos de Servicios que pueden ser movidos de un nodo a otro. Otra manera es hacer que diferentes instancias de la misma aplicación se ejecuten en múltiples nodos, de esta forma los usuarios solo tienen que reconectarse al nodo alternativo. Después de eliminar los Puntos Simples de Fallo, es importante hacer que las aplicaciones operen correctamente en un ambiente de Alta Disponibilidad. Es decir, que las aplicaciones deben ser creadas o modificadas para que puedan trabajar en un *cluster*, éstas aplicaciones deberán tener las características ilustradas en la Tabla 1.22:

Habilidad de *failover* en otro nodo
Habilidad para reiniciarse
Soportar las conexiones de los usuarios en cualquier nodo y no en uno en particular
Funciones de monitoreo para determinar cuando una aplicación está ejecutándose o no
Bien definidos los procedimientos de inicio y paro de la aplicación
Bien definidos los procedimientos de respaldo, restauración y actualización

Tabla 1.22 Características deseables de las aplicaciones en un entorno de Alta Disponibilidad

1.7.12 IMPLANTACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD

El resultado de eliminar Puntos Simples de Fallo en los rubros de energía, discos, SPU, red y *software* es un verdadero *Cluster* de Alta Disponibilidad (Ver Figura 1.17).

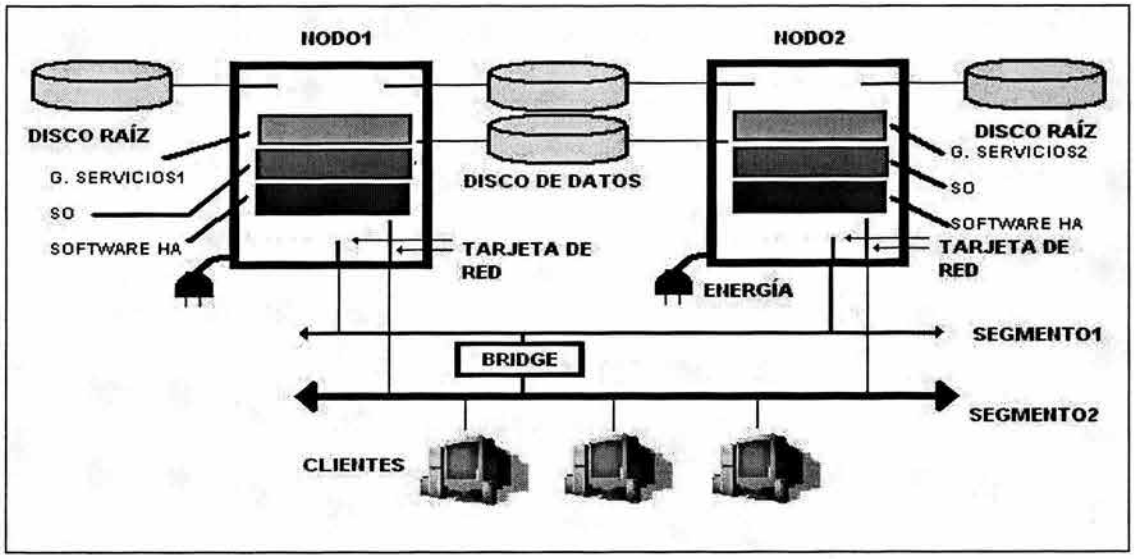

Figura 1.17 Configuración de un *cluster* de Alta Disponibilidad de dos nodos

En la Figura 1.17 podemos apreciar dos nodos trabajando en dos subredes LAN y discos raíz y de datos replicados donde las aplicaciones se ejecutan como parte de un Grupo de Servicio en cada nodo; y si existe un fallo de uno de los componentes de uno de los nodos, entonces el Grupo de Servicio puede ejecutarse en el otro nodo.

1.7.13 LA SOLUCIÓN COMPLETA DE ALTA DISPONIBILIDAD

Para desarrollar una completa solución de Alta Disponibilidad, se requiere mantener la Alta Disponibilidad dentro de una jerarquía de niveles del sistema. Las fallas en todos los niveles deberán ser detectadas rápidamente y darles la respuesta inmediata. Al mismo tiempo, realizar un mantenimiento planeado en todos los niveles con un mínimo de interrupciones del servicio.

La jerarquía de niveles se muestra en la siguiente Tabla 1.23:

NIVEL	CÓMO ES ALCANZADA LA ALTA DISPONIBILIDAD
A nivel *cluster*	La comunicación entre nodos debe ser altamente disponible. Los datos protegidos. Habrá múltiples nodos capaces de ejecutar los servicios
A nivel servidor	SPU debe ser redundante
A nivel sistema operativo	La replicación del *software* de sistema debe ser implantado
A nivel de administración de sistema y red	La administración distribuida del sistema y las herramientas de monitoreo de red deben ser altamente disponibles
A nivel de transacción de procesos	Todos los servicios deben ser altamente disponibles
A nivel manejador de base de datos	El manejador de base de datos debe ser capaz de iniciar en diferente nodo o ejecutarse en más de uno
A nivel aplicación	La aplicaciones deben ser robustas y capaces de recuperarse de ante errores.
A nivel *firmware*	La corrección de errores debe ser incorporado
A nivel *hardware*	Técnicas de intercambio deben ser incorporadas

Tabla 1.23 Jerarquía de niveles del Sistema de Alta Disponibilidad

1.8 SUMARIO

Con base en los elementos expuestos en páginas anteriores, podemos definir los pasos necesarios para diseñar cuidadosamente un Sistema de Alta Disponibilidad.

Estos pasos deberán ser llevados a cabo en el orden específico, de acuerdo a la Tabla 1.24:

PASO	DESCRIPCIÓN
1	Definir una meta, incluyendo una lista detallada de los objetivos del nivel de servicio para cada aplicación o servicio
2	Determinar la duración máxima de una interrupción aceptable
3	Medir la disponibilidad del sistema actual si es que está en uso. Esto incluye entender las estadísticas actuales sobre disponibilidad, incluyendo interrupciones planeadas o no planeadas. Estar seguro de las mediciones y lo que esto significa. Identificar todos los Puntos Simples de Fallo en el sistema actual
4	Evaluar las aplicaciones. ¿Cuáles son las mejoras o cambios deseados? ¿Cuál es el costo involucrado?
5	Diseñar o seleccionar la Arquitectura de Alta Disponibilidad que más se adecue a las necesidades
6	Adquisición y configuración de los componentes hardware-software, así como el soporte técnico
7	Crear o modificar el software de aplicación
8	Escoger las herramientas de administración adecuadas
9	Diseñar los procedimientos que serán seguidos cuando un fallo ocurra
10	Documentar estos procedimientos
11	Capacitar a los administradores y operadores sobre estos procedimientos
12	Revisar los procedimientos existentes
13	Documentar y registrar el estado actual del sistema

Tabla 1.24 Plan de diseño para un Sistema de Alta Disponibilidad

CAPÍTULO 2

Esquema de Seguridad para el Servidor Web

CAPÍTULO 2: ESQUEMA DE SEGURIDAD PARA EL SERVIDOR WEB

2.1 INTRODUCCIÓN

IMPORTANTE: Por cuestiones de seguridad, he omitido y cambiado algunos datos que se consideran confidenciales por parte de la Empresa.

2.1.1 IDENTIFICACIÓN DEL ESQUEMA DE SEGURIDAD

Título	Servidor Web de la Facultad de Ingeniería. Esquema de Seguridad
Autor	José de Jesús Ramírez Pichardo
Versión *Common Criteria*	CC versión 2.2
Versión del Esquema de Seguridad	Versión 1.0, julio de 2004

Tabla 2.1 Identificación del Esquema de Seguridad

2.2 ENTORNO DE SEGURIDAD

2.2.1 MISIÓN DE LA EMPRESA Y OBJETO DE EVALUACIÓN

Para realizar el Esquema de Seguridad del sitio Web de la Facultad de Ingeniería, haremos las siguientes consideraciones: trataremos a la Facultad de Ingeniería como una empresa, además consideraremos como Objeto de Evaluación al servidor que aloja, entre otros sitios, el sitio Web de la Facultad de Ingeniería.

Por tanto definimos lo siguiente (ver Tabla 2.2):

EMPRESA	FACULTAD DE INGENIERÍA
OBJETO DE EVALUACIÓN	SERVIDOR WEB

Tabla 2.2 Definición de la empresa y del Objeto de Evaluación

2.2.1.1 LA FACULTAD DE INGENIERÍA COMO EMPRESA

La Facultad de Ingeniería como toda organización o empresa está compuesta por varios elementos, tales como secretarías, divisiones, coordinaciones, entre otros. El organigrama de la empresa se ilustra en la Figura 2.1:

Figura 2.1 Organigrama de la institución

2.2.1.2 MISIÓN DE LA EMPRESA

La misión de la empresa, Facultad de Ingeniería, es la siguiente:

"Formar integralmente recursos humanos en los niveles de licenciatura, especialidad y posgrado, para que sean competitivos en el ámbito nacional e internacional como ingenieros de la más alta calidad; con habilidades y actitudes que les permitan el mejor desempeño en el ejercicio profesional, la investigación y la docencia; con capacidad para aprender durante toda la vida y mantenerse actualizados en los conocimientos de vanguardia; con una formación humanista que sustente sus actos y sus compromisos con la Universidad y con México, para que coadyuven al mejoramiento social, económico, político y cultural de la nación."[1]

[1] Del Sitio www.ingenieria.unam.mx

Dentro de esta empresa, la Dirección delegó la responsabilidad del Objeto de Evaluación a dos Secretarías: Secretaría General y Secretaría de Servicios Académicos. Como veremos, el responsable directo del Objeto de Evaluación es un organismo que depende directamente de Secretaría General: la Unidad de Servicios de Cómputo Académico. Por tal motivo creo necesario conocer estos organismos.

La Secretaría de Servicios Académicos *"tiene como objetivo coordinar la ejecución de las actividades de administración escolar, del programa editorial de la Facultad, de apoyo a maestros y alumnos y, en general, proporcionar los servicios que se requieran en la Facultad para el mejor desempeño de las actividades académicas. Las funciones del secretario de servicios académicos son entre otras: proporcionar a profesores y alumnos los apoyos de administración escolar, difusión, edición de materiales didácticos, bibliotecas, fotocopiado, audiovisuales, realizar los estudios académicos y preparar los informes que le sean solicitados por la dirección, y que coadyuven al desarrollo académico de la Facultad; coordinar y supervisar los servicios de transporte y mantenimiento que se requieren en la Facultad, coordinar y promover aquellas actividades socioculturales, deportivas y recreativas que propicien una formación integral de los estudiantes de la Facultad."* [2]

La Secretaría General *"tiene como objeto planificar, implantar, evaluar y coordinar la ejecución de los asuntos con carácter académico de la Facultad"* [2]. Las funciones del Secretario General, entre otras son: ejercer el secretariado del Consejo Técnico, supervisar las actividades de los órganos de apoyo académico a su cargo, proporcionar a las divisiones de la Facultad el apoyo que le soliciten en lo relativo a servicios de información, de comunicación, de cómputo y otros de su competencia, solicitar a las unidades a su cargo la formalización de estudios, investigaciones e informes que requiera el desarrollo académico de la Institución, colaborar con el director en las actividades de planeación, evaluación y apoyo académico de la Facultad. Para llevar a cabo estas funciones, el secretario general se apoya en los subórganos a su cargo (Ver Tabla 2.3):

Unidad de Servicios de Cómputo Académico (UNICA)
Coordinación de Comunicación
Coordinación de Programas de Atención Diferenciada para Alumnos
Coordinación de Superación del Personal Académico
Departamento de Información y Estadística
Unidad de Apoyo al Consejo Técnico
Departamento del Personal Académico
Unidad de Apoyo Editorial
Centro de Docencia

Tabla 2.3 Subórganos de la Secretaría General

[2] Del Sitio www.ingenieria.unam.mx

"La Unidad de Servicios de Cómputo Académico es una dependencia de la Secretaría General de la Facultad de Ingeniería, cuya finalidad principal es la de proporcionar, en el ámbito institucional, los servicios de apoyo en cómputo que la comunidad de la Facultad requiere, recursos de cómputo comerciales y de alta especialización que el avance de la educación, el desarrollo de la informática y el ejercicio profesional demanden." [3]

La Unidad de Servicios de Cómputo Académico se subdivide en (Ver Figura 2.2):

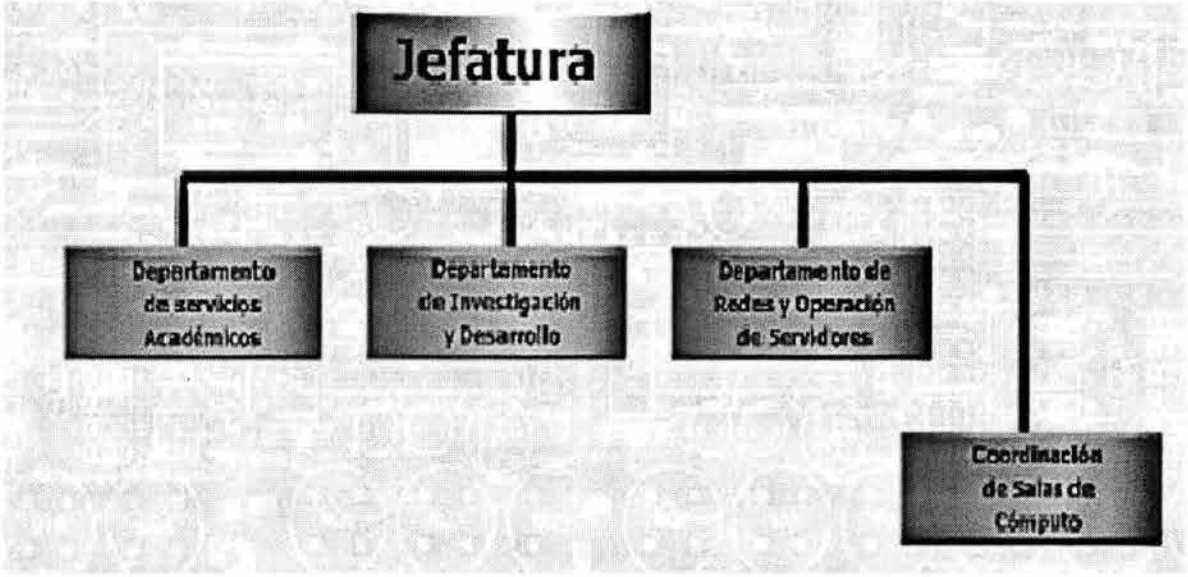

Figura 2.2 Organigrama de la Unidad de Servicios de Cómputo Académico

El departamento de Redes y Operación de Servidores tiene a su cargo (físicamente) el Objeto de Evaluación. El Departamento de Redes y Operación de Servidores es *"responsable de la administración, operación, mantenimiento y seguridad de la red de comunicación de la Facultad y de la intercomunicación con la red central de la UNAM; así como del desarrollo e implementación de proyectos para la expansión del servicio. Administra y da mantenimiento a las cuentas de servicios de correo electrónico y de Internet de la Facultad. Unas 6,000 cuentas individuales, en promedio anual."* [3]

Por tanto, éste departamento tiene, para con el Objeto de Evaluación, las siguientes responsabilidades (Ver Tabla 2.4):

Administración
Operación
Mantenimiento
Seguridad

Tabla 2.4 Responsabilidades del Departamento de Redes y operación de Servidores en relación al Objeto de Evaluación

[3] Del Sitio www.ingenieria.unam.mx/~unica

2.2.1.3 MISIÓN DEL OBJETO DE EVALUACIÓN

La Facultad de Ingeniería, en respuesta a la creciente globalización de las Tecnologías de la Información, cuenta con su propio sitio: *www.ingenieria.unam.mx*

La misión del Objeto de Evaluación es: dar hospedaje o *web hosting* a la página de la Facultad de Ingeniería, a sitios Web de organismos internos y de personalidades que forman parte de la comunidad de la Facultad de Ingeniería. Además pone a disposición todos los recursos necesarios para la ejecución de sistemas que son de vital importancia para la comunidad.

El Objeto de Evaluación del Esquema de Seguridad es, como ya he definido, el Servidor Web. En la Tabla 2.5 lo identificamos:

NOMBRE CANÓNICO	cozumel.fi-a.unam.mx
ALIAS O SOBRENOMBRES	www.ingenieria.unam.mx www.fi-a.unam.mx ingenieria.unam.mx
IP	192.168.0.1

Tabla 2.5 Identificación en la red del Objeto de Evaluación

2.2.2 ENTORNO FÍSICO

2.2.2.1 ÁREA PERIMETRAL

El Objeto de Evaluación se encuentra ubicado en el Ala Norte o Principal de la Facultad de Ingeniería, dentro de las instalaciones de la Unidad de Servicios de Cómputo Académico. En la figura 2.3 podemos hacer un seguimiento general de la ubicación:

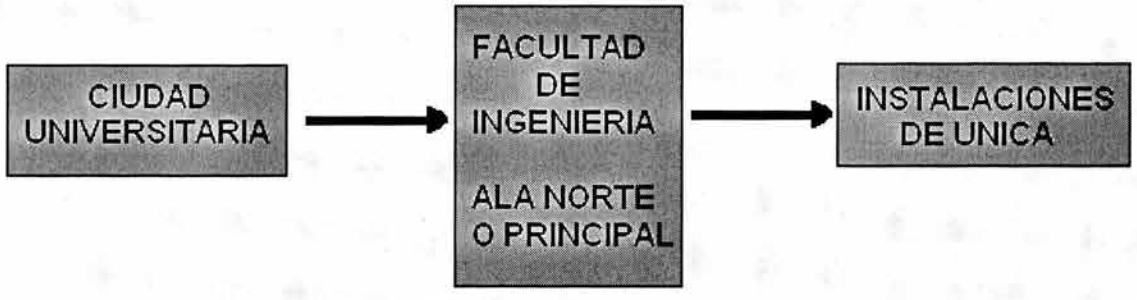

Figura 2.3 Ubicación del Objeto de Evaluación (1)

En el siguiente esquema podemos ubicar las instalaciones de UNICA en el Ala Norte o Principal de manera más detallada (Ver Figura 2.4):

Figura 2.4 Ubicación del Objeto de Evaluación (2)

Las instalaciones de UNICA están ubicadas en el edificio que pertenece a Secretaría General y USECAD, este es un corte transversal del edificio (Ver Figura 2.5):

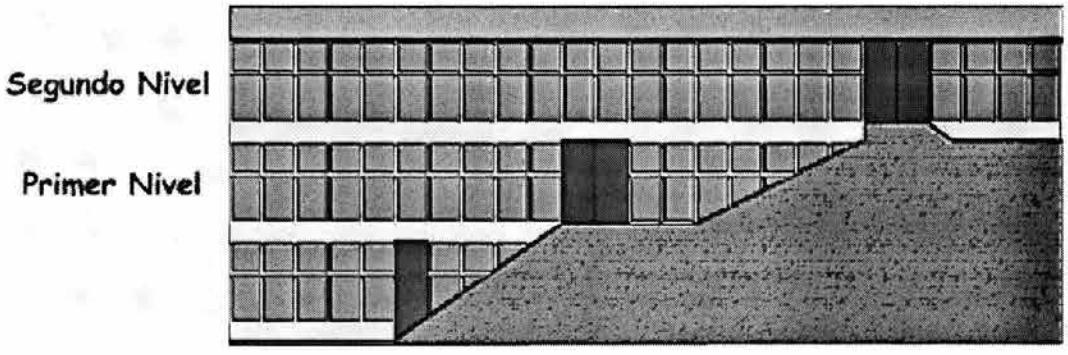

Figura 2.5 Ubicación del Objeto de Evaluación (3)

El Objeto de Evaluación está en el Primer Nivel, donde actualmente se encuentran las salas de cómputo, Control de Salas, Cuarto de Servidores, el cubículo del Área de Seguridad, la Bodega, el cubículo del Área de Administración UNIX, el cubículo del Jefe del Departamento de Redes y Operación de Servidores, y la Jefatura de UNICA (Ver Figura 2.6).

1. Jefe del Depto. de Redes y Operación de Servidores
2. Área de Administración UNIX
3. Área de Seguridad

Figura 2.6 Ubicación del Objeto de Evaluación (4)

Dentro del Cuarto de Servidores se encuentra físicamente el Objeto de Evaluación.

2.2.2.2 CUARTO DE SERVIDORES

El Cuarto de Servidores tiene un área de 20 m^2 (4x5 m), la pared frontal y una lateral (la que da a Control de Salas) tienen ventanales cubiertos con cortinas que no son translúcidas. La puerta de acceso, es de tamaño estándar, de madera con dos chapas sencillas, una de pasador y otra de picaporte. No cuenta con un sistema de ventilación, la iluminación es suficiente.

Dentro de este Cuarto de Servidores, se encuentran 4 equipos PC, dos UPS, dos mesas, dos estantes.

Los equipos PC corresponden a los servidores que describo a continuación (Ver Tabla 2.6):

SERVIDOR	DESCRIPCIÓN
Mail Server	Servidor de Correo Electrónico de la Facultad de Ingeniería.
Web Server	Servidor Web de la Facultad de Ingeniería
Application Server	Servidor de Aplicaciones para uso académico
Proxy Server	Servidor *proxy*, que da salida a Internet a los equipos de las Salas de Cómputo. *Log Server*, en el cual se almacenan las bitácoras de los servidores críticos.

Tabla 2.6 Equipos en el Cuarto de Servidores

El Objeto de Evaluación está embebido en una reja de varilla empotrada al suelo, no tiene candado. (Ver Figura 2.7):

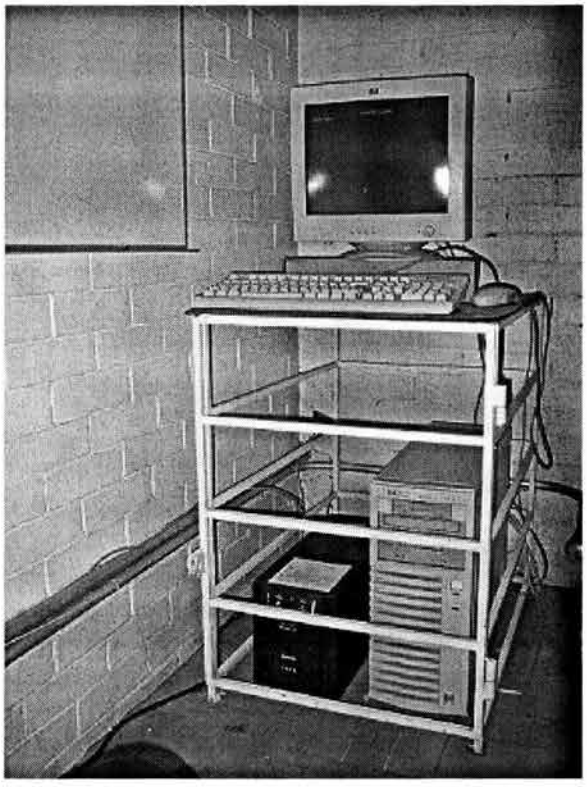

Figura 2.7 Objeto de Evaluación

El acceso al Cuarto de Servidores está restringido solo a Administradores de Servidores pertenecientes al Área de Administración UNIX. (Ver Figura 2.8)

Figura 2.8 Acceso sólo a personal autorizado

Las copias de seguridad o los respaldos están ubicados en este lugar y habrá que robustecer el control estricto sobre su manejo.

2.2.2.3 OBJETO DE EVALUACIÓN

El servidor es de la marca *Hewlett Packard*, modelo NETSERVER E200 HP, con las siguientes características (ver Tabla 2.7):

Característica	Descripción
Modelo	Netserver E200
Procesador	*Pentium III*
Velocidad	667 MHz
Bus	133 MHz
Caché	L2 - 256KB
RAM	320 MB
Disco Duro	9.1 GB SCSI
Tarjeta de Red	10/100 TX
Controladora discos	SCSI
Unidades de almacenamiento externo	CD-ROM 40X *Floppy Disk* Unidad de cinta
Controladora de video	4 MB RAM
Monitor	HP de 17 pulgadas
Teclado	HP
Mouse	HP

Tabla 2.7 Características físicas del Objeto de Evaluación

Cuenta con una unidad UPS SMART-UPS 1500 con seis contactos que provee energía de respaldo cercano a 60 minutos de operación.

2.2.2.4 LA CONEXIÓN A INTERNET

Los nodos o puntos de red que emplean los servidores para ofrecer sus servicios a Internet, no están canalizados y por tanto no están protegidos. Este cableado sale del Cuarto de Servidores, pasa por el Área de Seguridad, cruza las Salas de Cómputo y llega a la Bodega donde se encuentran los equipos activos (*switches*, entre otros). Esta bodega está resguardada bajo llave, sin embargo gente ajena a la administración del Objeto de Evaluación tiene acceso. La red es *Ethernet* a 10/100 Mbps, el *hardware* que está involucrado es cableado de par trenzado o UTP categoría 5, conectores RJ-45, *switches, hubs*. En la Figura 2.9 podemos observar la forma en que el Objeto de Evaluación se conecta a Internet.

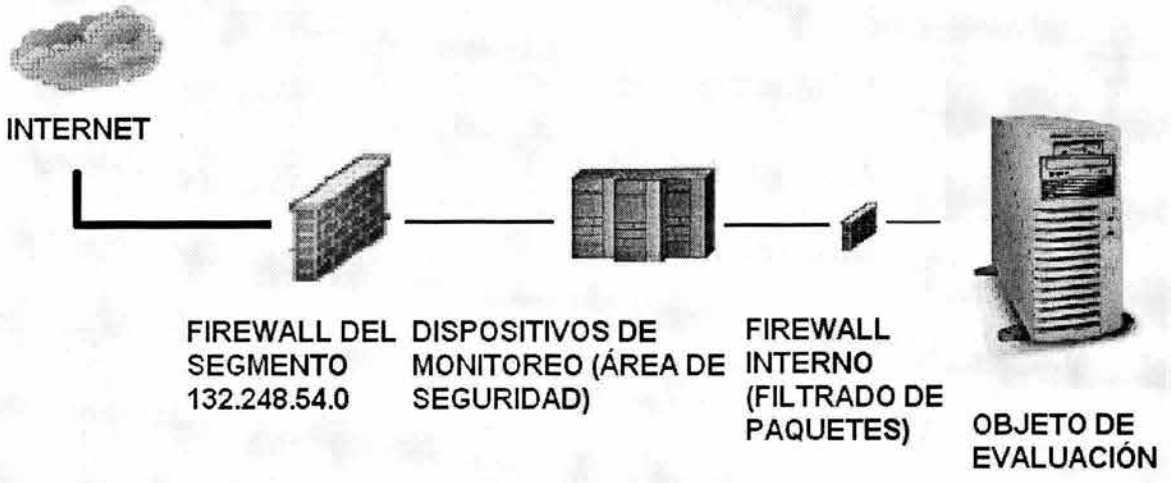

Figura 2.9 El Objeto de Evaluación e Internet

Básicamente para que el Objeto de Evaluación interactué con Internet, existen tres barreras: la primera de ellas es un *Firewall* que protege parte del segmento de red 132.248.54.0, posteriormente el Área de Seguridad de UNICA implantó un conjunto de Dispositivos de Monitoreo. Una vez pasadas estás dos barreras, llegamos al *Firewall* interno que está instalado en el propio servidor a nivel de *software*, el cual filtra los paquetes de equipos del segmento 132.248.54.0, los cuales están atrás del *Firewall*.

2.2.3 ENTORNO LÓGICO

2.2.3.1 SISTEMA OPERATIVO

El sistema operativo que administra los recursos del Objeto de Evaluación es *Linux Fedora Core 1*, cuya versión del *kernel* o núcleo de sistema es *2.4.22-1.2115.nptl.* Sobre éste aspecto la Tabla 2.8 describe las características analizadas:

La instalación fue personalizada, es decir se eliminó todo *software* que no tiene relación con la función que desempeña el Objeto de Evaluación
Los puertos TCP y UDP innecesarios están cerrados
Servicios innecesarios están desactivamos (Sendmail, NFS, NIS, SNMP, entre otros)
Están instaladas las actualizaciones o parches, pero no las actuales
Se cambiaron los *banners* de bienvenida los cuales mostraban información sobre el sistema
Todo usuario del sistema tiene asignado un password "seguro", y las cuentas de sistema han sido bloqueadas
Tiene el proceso de *syslog* corriendo, es decir, esté proceso se encarga de la generación de bitácoras del sistema
La ejecución de compiladores y comandos de administración está restringida
Los recursos como son memoria RAM, procesos, No. de archivos o inodos está restringida
Tiene corriendo el demonio calendarizador de tareas (cron)
Los sistemas de archivos de los usuarios */home, /users, /usuarios* tienen establecido el sistema de cuotas

Tabla 2.8 Estado del sistema operativo

El disco duro está particionado según la Tabla 2.9:

PARTICIÓN	TAMAÑO (%)
swap	7.11
/boot	1.11
/casa	22.22
/home	6.62
/users	4.44
/usuarios	22.22
/tmp	2.18
/usr	19.96
/var	4.44
/	9.7

Tabla 2.9 Particiones del disco duro

En la partición */casa* se encuentran ubicados los archivos del sitio principal de la Facultad de Ingeniería (*index.php*, entre otros). En */home, /users y /usuarios* están los *home directory* de los usuarios del sistema. La partición */usr* está destinada para la instalación de la aplicaciones como Apache, PHP y PostgreSQL.

En cuanto a tareas programadas (cron), el Objeto de Evaluación tiene las siguientes tareas (Ver Tabla 2.10):

*/5 * * * * /root/MirrorServer/bin/serv.bash
0 * * * 1-5 /bin/sh /usr/local/doc/logcheck/logcheck.sh
0 13,23 * * 6,0 /bin/sh /usr/local/doc/logcheck/logcheck.sh
*/5 * * * 1-5 /root/MirrorServer/bin/sistema_acad.bash
*/5 9-20 * * 1-5 /root/MirrorServer/bin/usuarios_altasystem.bash

Tabla 2.10 Tareas programadas en el Objeto de Evaluación

La primera tarea verifica que los servicios de Apache y PostgreSQL estén siempre en ejecución, la siguientes dos tareas realizan un análisis de la bitácoras generadas por el sistema. Las últimas dos tareas envían información a dos sistemas que se ejecutan en el servidor cancun.fi-a.unam.mx.

2.2.3.2 SOFTWARE

El Objeto de Evaluación tiene instalado el servidor HTTP Apache, el servidor de base de datos PostgreSQL, y el lenguaje de programación PHP (Ver Tabla 2.11):

SOFTWARE	VERSIÓN	DESCRIPCIÓN
Apache Web Server	x.y.z	Servidor Web HTTP/HTTPS de código fuente libre
PHP	x.y.z	Lenguaje de programación de *scripting* de código fuente libre
OpenSSL	x.y.z	Librerías y herramientas para asegurar la confidencialidad de la información Web
PostgreSQL	x.y.z	Servidor de base de datos de código fuente libre

Tabla 2.11 Software instalado en el Objeto de Evaluación

El Objeto de Evaluación tiene instalado SSL (*Secure Socket Layer*) que permite implantar métodos criptográficos en el transporte de la información entre cliente y Servidor Web. Esta capacidad no está activada por razones fuera del ámbito técnico. Este cambio de HTTP a HTTPS provocaría un cambio en la forma de conexión de los clientes hacia el servidor, de http://www.ingenieria.unam.mx a https://www.ingenieria.unam.mx. Los administradores, para evitar este cambio radical, actualmente investigan métodos de redireccionamiento de puertos.

2.2.3.2.1 CONFIGURACIÓN DE APACHE

En general se considera correcta la configuración de Apache, en la Tabla 2.12 se ilustra parte de la configuración:

El usuario dueño de los procesos de Apache es *webadmin* y no *root* o *nobody*
El *DocumentRoot* es */casa/httpd/html* y se ubica en una partición diferente de /
La política para el acceso al sistema de archivos desde la Web, es Prohibitiva, es decir, se prohíbe todo y posteriormente se permite sólo a ciertos directorios, de esta forma el acceso a / está prohibido
En *DocumentRoot* y en los directorios WWW de los usuarios no se permite la ejecución de CGI's, no se puede exhibir el contenido del directorio (*indexes*), sólo se da seguimiento a las ligas o vínculos que pertenecen a los usuarios (de esta forma no se puede observar el */etc/passwd* desde el Web, por ejemplo), no se permite la edición de archivos *.htaccess* con los cuales se configuran ciertas características del servidor, se limita el uso de todos lo métodos excepto GET Y POST, empleados por los *scripts* de PHP y HTML
El directorio WWW de *root* está inhabilitado
Se tiene acceso a ciertos directorios, pero sólo por máquinas a las cuales les es permitido por el Servidor Web
No se tiene acceso a archivos .ht*, .inc*, *.bak, *.old, *.sql, *.SQL, *.BAK, *.OLD por el contenido crítico que contienen
La directiva *ServerTokens* está en *Prod*, es decir que el *banner* del producto que muestra en ciertas conexiones es "Apache", de otra forma mostraría información sobre la versión, módulos compilados y el sistema operativo huésped
El manual fue eliminado
Genera bitácoras de actividad del Servidor Web
Se configuró sólo un directorio en el cual se pueden ejecutar CGI's, pero sólo un usuario puede hacerlo, de está forma se mejora la administración y seguridad de CGI's

Tabla 2.12 Parte de la configuración de Apache

2.2.3.2.2 CONFIGURACIÓN DE POSTGRESQL

En general, el servidor de base de datos está correctamente configurado, en la Tabla 2.13 se muestra parte de la configuración:

El dueño de los procesos de PostgreSQL es *postgres* y no *root*
La base de datos administradora (*Template1*) está protegida por *password* y sólo el usuario *postgres* puede tener acceso
La política de acceso es Prohibitiva, se prohíbe el acceso a todo mundo excepto a ciertas máquinas de dos segmentos de la red local
Todas las bases de datos cuentan con *password* y además sólo se asigna su uso a usuarios definidos
Genera una bitácora de actividad del servidor de base de datos

Tabla 2.13 Parte de la configuración de PostgreSQL

El *software* está correctamente configurado y personalizado, sin embargo, el servidor no cuenta con la última actualización.

2.2.3.3 SOFTWARE DE SEGURIDAD

La Tabla 2.14 describe el *software* de seguridad instalado en el Objeto de Evaluación:

SOFTWARE	VERSIÓN	DESCRIPCIÓN
SCANLOG	x.y.z	Es un monitor de barrido de puertos, es decir, detecta e informa vía *log* de un barrido de puertos de un *host* hostil
PORTSENTRY	x.y.z	Detecta una conexión a un puerto no válido o un barrido de puertos y toma acciones de bloqueo contra el *host* hostil
IPTABLES	x.y.z	Permite el filtrado de paquetes tanto de entrada como de salida, con está herramienta se implantó el *Firewall* interno
TCP_WRAPPERS	x.y.z	Es un monitor de paquetes, que trabaja con los archivos */etc/hots.allow y /etc/hosts.deny*. Máquinas que se encuentren el */etc/hosts.deny* no pueden realizar ciertas conexiones y las incluidas en */etc/hosts.allow* si pueden hacerlo
OPENSSH	x.y.z	Permite establecer conexiones remotas de manera segura puesto que la información viaja de manera cifrada entre cliente-servidor
WEBALIZER	x.y.z	Es un analizador de bitácoras Web, que genera informes y estadísticas sobre la actividad del servidor.
LOGCHECK	x.y.z	Analiza las bitácoras generadas por el sistema, eliminando toda información "no crítica", y elaborando un reporte de lo que si considera crítico

Tabla 2.14 Software de seguridad instalado

El *software* de seguridad está correctamente configurado, sin embargo, el servidor no cuenta con el *software* de seguridad actualizado.

2.2.3.4 SISTEMAS Y APLICACIONES

En el Objeto de Evaluación se ejecutan los siguientes sistemas (Ver Tabla 2.15):

SISTEMA	DESCRIPCIÓN
altasystem (www.ingenieria.unam.mx/inscripciones)	Sistema de Inscripciones a las Salas de UNICA, éste sistema da de alta usuarios, hace uso de una base de datos local y envía los datos al servidor cancun.fi-a.unam.mx, *Application Server* y *BD Server* para generar sus cuentas. Sólo ciertas máquinas pueden acceder a este sistema.
correocfi (www.ingenieria.unam.mx/~correocfi)	Sistema de Correo para Académicos, este sistema permite que los académicos puedan hacer un preregistro para obtener una cuenta de correo en el *Mail Server*. Todo mundo tiene acceso.
correocfiadmin (www.ingenieria.unam.mx/~correocfiadmin)	Interfaz del Administrador del Sistema de Correo para Académicos, con este sistema se finaliza el registro de los académicos para la obtención de su cuenta de correo, envía los datos necesarios a *Mail Server*. Todo mundo tiene acceso, sin embargo sólo ciertas máquinas debieran tener acceso.
estadisticas (www.ingenieria.unam.mx/estadisticas)	Este directorio contiene las estadísticas que genera Webalizer, y sólo el administrador tiene acceso.
scosu	Este sistema no está en el objeto de evaluación, sólo cuenta con una base de datos llamada *scosu*. Sólo tienen acceso a está base de datos dos segmentos de la red local. Sin embargo sólo debieran tener acceso ciertos equipos.

Tabla 2.15 Sistemas y aplicaciones relacionadas con el Objeto de Evaluación

Para llevar a cabo algunas tareas el Objeto de Evaluación establece mecanismos de confianza con *Mail Server, Application Server, BD Server y Proxy Server.*

Para ello emplea SSH para permitir que *root* pueda conectarse a estos equipos sin que tenga que ingresar un *password.* (Ver Figura 2.10)

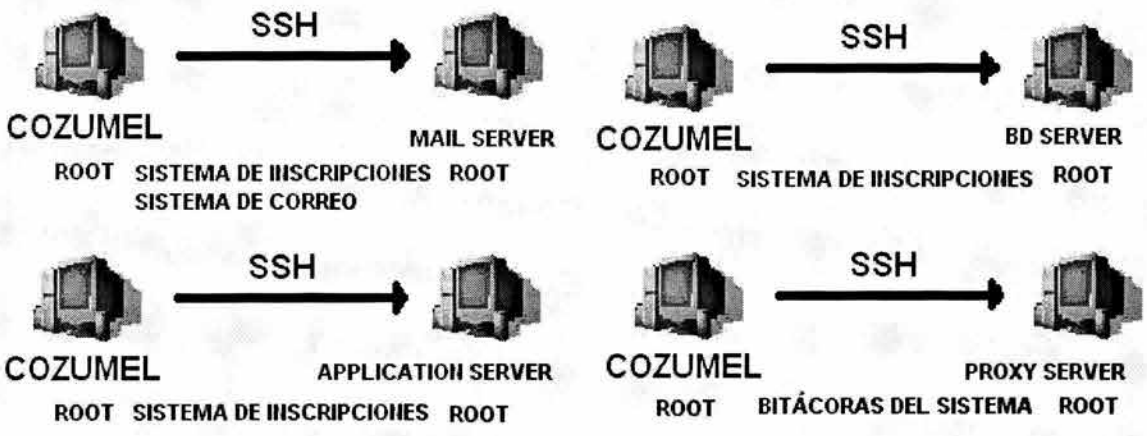

Figura 2.10 Equipos y relaciones de confianza

2.2.4 USUARIOS

2.2.4.1 TIPOS DE USUARIO

Con respecto al Objeto de Evaluación definimos dos tipos de usuarios (Ver Tabla 2.16):

USUARIO	DESCRIPCIÓN
Usuarios WWW	Realizan conexiones mediante navegador
Usuarios del sistema	Usuarios que mantienen sus desarrollos y sitios en el servidor, se conectan vía SSH, navegador o realizan una conexión a su base de datos
Administradores	Todo personal autorizado para llevar a cabo las tareas de administración, operación, mantenimiento y seguridad

Tabla 2.16 Tipos de usuario

Tanto los Usuarios del sistema como los Administradores están plenamente identificados.

2.2.4.2 USUARIOS WWW

Son todas aquellas personas que visualizan e interactúan con el Objeto de Evaluación a través de un navegador gráfico o modo texto. No están identificados porque pueden ser usuarios de otras regiones.

2.2.4.3 USUARIOS DEL SISTEMA

Los Usuarios del sistema están divididos, de acuerdo a sus características comunes se cuentan con los siguientes grupos.

Grupo unicafi: Los usuarios de este grupo son parte del personal de la Unidad de Servicios de Cómputo Académico. Se alojan sistemas propios de dicha institución, así como las cuentas de los administradores del servidor.

Grupo trabajadoresadm: Los usuarios de este grupo son trabajadores y personal administrativo que laboran en la Facultad de Ingeniería.

Grupo administrativos: Los usuarios que pertenecen a este grupo son administrativos de mayor nivel en la Facultad de Ingeniería.

Grupo ayudprof: Sus usuarios son personal que colabora con los profesores de la Facultad de Ingeniería.

Grupo externos: Agrupa cuentas de usuario especiales, usuarios que no pertenecen a la Facultad de Ingeniería, por ejemplo alguna vez se dio alojamiento al sitio Web de otra Facultad del campus universitario.

Grupo facultad: Este grupo es el más numeroso, agrupa cuentas de usuario de personajes y organismos propios de la Facultad de Ingeniería.

Grupo funcionarios: A este grupo pertenecen cuentas de usuario que pertenecen a funcionarios de la Facultad de Ingeniería.

Grupo investigadores: Agrupa a los investigadores de la Facultad de Ingeniería.

Grupo profesores: Este grupo contempla a todos los profesores que tienen la necesidad de mantener un sitio Web en Internet.

Grupo proyecto: Como su nombre lo indica, este grupo engloba a todos los usuarios u organismos que mantienen un proyecto en la Facultad de Ingeniería.

Grupo tecacad: Este grupo engloba a todos los usuarios que sean Técnicos Académicos en la Facultad de Ingeniería.

En cuanto a los usuarios podemos hacer ciertas observaciones, existen cuentas que no tienen una persona u organismo responsable, otras con referencias no muy claras. Posiblemente existan cuentas que no se emplean ya. Todas las cuentas tienen *password* y cuentan con su respectiva cuota tanto de espacio en disco duro como de número de archivos permitidos.

2.2.4.4 ADMINISTRADORES

Es el personal del Área de Administración UNIX encargado de la administración, operación, mantenimiento y seguridad del Objeto de Evaluación y del resto de servidores y equipos a cargo del Departamento de Redes y Operación de Servidores. El grupo esta formado por 7 personas, las cuales tienen la contraseña de administrador del Objeto de Evaluación y del resto de los equipos. Solo hay una persona encargada plenamente y el resto la apoya. Todos tienen las contraseñas apuntadas en un papel que resguardan en sus bolsas, carteras, etc. En general no tienen las contraseñas memorizadas.

2.3 HIPÓTESIS

Las hipótesis son todas aquellas suposiciones que en conjunto nos indican el cómo debiera operar seguramente el entorno de estudio.

2.3.1 HIPÓTESIS GENERALES

H0 El Objeto de Evaluación opera las 24 horas del día, los 7 días de la semana, los 365 días del año

2.3.2 SOBRE EL CUARTO DE SERVIDORES

H1 Acceso sólo a personal autorizado

H2 Las llaves de acceso están en manos de los administradores y del jefe del departamento

H3 Los ventanales cuentan con protección metálica para evitar intrusiones físicas, además de cortinas que no permiten la visión hacia el interior

H4 La puerta de acceso es de un material resistente

H5 Cuenta con sistema de ventilación

H6 En caso de incendio existe un extintor capaz de sofocar el siniestro

H7 Cada equipo cuenta con su propia unidad UPS

H8 Está ordenado y limpio

H9 La rejilla donde está embebido el Objeto de Evaluación cuenta con candado, y la llave está en manos de los administradores y del jefe del departamento

2.3.3 SOBRE EL OBJETO DE EVALUACIÓN

H10 Cuenta con unidad UPS

H11 Cuenta con los recursos *hardware* requeridos, como son memoria RAM, tamaño del disco duro, procesador, entre otros

H12 Cuenta con al menos un disco duro de respaldo ante la falla del original (*disk mirroring*)

H13 Cuenta con una tarjeta de red adicional

H14 Existe un servidor *mirror* o réplica ante la falla del servidor principal

2.3.4 SOBRE LAS COPIAS DE SEGURIDAD O RESPALDOS

H15 Se realizan periódicamente y son debidamente revisados

H16 Están resguardados en un sitio lejos del Cuarto de Servidores y se lleva un control sobre ellos

2.3.5 SOBRE EL CABLEADO

H17 Todo el cableado desde el Cuarto de Servidores hasta la Bodega, donde se encuentra el equipo activo, está en sus correspondientes canaletas y protegido ante cualquier accidente intencionado o no. Se cuenta con redundancia en el cableado

2.3.6 SOBRE LA BODEGA

H18 Sólo personal autorizado puede acceder a la Bodega donde se ubica el equipo activo

2.3.7 SOBRE EL EQUIPO ACTIVO

H19 Se cuenta con redundancia en el equipo activo

2.3.8 SOBRE EL FIREWALL

H20 El *Firewall* que protege parte del segmento 132.248.54.0 cuenta con una unidad UPS y está debidamente asegurado

2.3.9 SOBRE EL SISTEMA OPERATIVO

H21 Está actualizado, es decir se le han aplicado las actualizaciones correspondientes

H22 Cuenta sólo con el *software* que se requiere

H23 Los puertos y servicios innecesarios están inhabilitados

H24 Los *banners* de bienvenida están editados de tal forma que no muestran información relativa al sistema operativo y *software* instalado

H25 La ejecución de compiladores y comandos de administración ha sido limitado

H26 El uso de recursos como son los procesos, memoria, No. de inodos, entre otros han sido limitados

H27 Los archivos SETUID y SETGID tienen los permisos correctos

H28 El uso del demonio calendarizador de tareas (cron) has sido restringido

H29 Las tareas programadas están ejecutándose adecuadamente

H30 Las particiones */home, /users y /usuarios* tienen activado el sistema de cuotas

H31 Se están generando las bitácoras del sistema, es decir el demonio de *syslog* está ejecutándose

2.3.10 SOBRE LOS USUARIOS

H32 Las cuentas especiales con *daemon, bin, sys*, entre otras, están bloqueadas

H33 Todas las cuentas de usuario tienen contraseñas fuertes

H34 Las cuentas de usuarios innecesarias han sido eliminadas

H35 Todos usuario tiene asignada su respectiva cuota en cuanto a espacio en disco duro y No. de archivos o inodos permitidos

H36 Los permisos de los usuarios son los adecuados para evitar el fisgoneo, robo o modificación de su información

H37 Las cuentas de usuario están debidamente configuradas y tienen asignado un responsable del cual se tiene ubicación en caso de emergencia

2.3.11 SOBRE EL SOFTWARE

H38 Se han instalado las últimas versiones de Apache, PHP y PostgreSQL

2.3.12 SOBRE APACHE

H39 El usuario dueño de los procesos de apache es *webadmin* y no *root* o *nobody*

H40 El directorio */casa*, donde se encuentra el sitio principal, está ubicado en una partición independiente de / y del directorio de instalación del Servidor Web

H41 La política de acceso al sistema de archivos desde Internet es Prohibitiva

H42 Solo se pueden ejecutar CGI's desde un directorio especial

H43 No se puede observar el contenido de directorios desde Internet

H44 Desde el navegador se pueden seguir sólo los vínculos que sean propiedad del usuario

H45 Esta restringido el uso de archivos *.htaccess*

H46 A excepción de los métodos GET y POST, todos los métodos están limitados

H47 El directorio WWW de *root* está inhabilitado

H48 Los sitios que sólo pueden ser visualizados por personal autorizado están restringidos

H49 No se tiene acceso a los archivos *.inc, .ht*, *.bak, *.old, *.sql, *.SQL, *.OLD, *.BAK

H50 La directiva *ServerTokens* tiene el valor de *Prod*, evitando la publicación de información no deseada

H51 El manual *online* fue eliminado

H52 El Servidor Web genera bitácoras sobre su actividad, las cuales son analizadas adecuadamente por los administradores

H53 Se implementa el mecanismo SSL/TLS para cifrar el transporte por el cual viajan tanto las solicitudes de los clientes Web y las respuestas del Objeto de Evaluación

H54 Se limitan los recursos que emplea Apache para evitar una subalimentación

H55 Los CGI's que ya no son utilizados han sido retirados

2.3.13 SOBRE POSTGRESQL

H56 Los procesos de PostgreSQL son ejecutados por el usuario *postgres* y no *root* o *nobody*

H57 La base de datos *Template1* está protegida por *password* y sólo el usuario *postgres* puede acceder a ella

H58 La política de acceso a las bases de datos es Prohibitiva

H59 Todas las bases de datos están protegidas con contraseña y sólo su propietario tiene acceso a ella

H60 El servidor de base de datos genera una bitácora de actividad que analizan los administradores frecuentemente

2.3.14 SOBRE EL SOFTWARE DE SEGURIDAD

H61 Todo el *software* de seguridad está actualizado y correctamente configurado

H62 Se tienen instalados y configurados dos sistemas detectores de barridos de puertos y ante esto se toman acciones automáticas para proteger el Objeto de Evaluación

H63 Se tiene configurado un *Firewall* interno.

H64 Se tiene instalado TCP_WAPPERS para controlar el acceso remoto al Objeto de Evaluación

H65 Cuenta con OpenSSH para realizar conexiones remotas seguras hacia y desde el Objeto de Evaluación

H66 Para el análisis de las bitácoras Web, se cuenta con Webalizer

H67 Para el análisis de las bitácoras del sistema se cuenta con la herramienta Logcheck

H68 Cuenta con un sistema detector de intrusos

H69 Cuenta con una herramienta para asegurar la integridad de los archivos de configuración del servidor

2.3.15 SOBRE LOS SISTEMAS Y APLICACIONES

H70 El control de acceso a los sistemas y aplicaciones que se ejecutan en el Objeto de Evaluación está limitado si así lo requieren

H71 Los mecanismos de confianza establecidos entre el Objeto de Evaluación y otros servidores están correctamente configurados y asegurados

H72 La programación de los sistemas y aplicaciones contemplan la inyección de código arbitrario, emplean una filosofía de programación de código seguro

2.3.16 SOBRE LOS ADMINISTRADORES

H73 Los administradores memorizan las contraseñas y evitan apuntarlas en papeles que puedan ser extraviados o revisados por otras personas

H74 Revisan frecuentemente todas las bitácoras generadas, tanto del sistema como de Apache y PostgreSQL

H75 Realizan respaldos periódicamente y revisan que estén en correcto estado

H76 Realizan un monitoreo constante del sistema, procesos, conexiones, la actividad de los usuarios, entre otros

H77 Existe más de un administrador capacitado para realizar las tareas de administración, operación, mantenimiento y seguridad del Objeto de Evaluación

H78 Llevan acabo campañas que informan a los usuarios sobre lo que es la ingeniería social y la amenaza que representa

H79 Realizan frecuentemente análisis sobre el sistema operativo, *software*, sistemas y aplicaciones en busca de vulnerabilidades

H80 Están en constante capacitación y buscan estar siempre informados sobre las nuevas amenazas y vulnerabilidades

H81 Ejecutan herramientas para encontrar contraseñas débiles de los usuarios

2.4 AMENAZAS

2.4.1 AMENAZAS FÍSICAS

A1 Ausencia por enfermedad de las personas encargadas de la administración del servidor

A2 Problemas laborales de la institución

A3 El acceso a la cuenta *root* o de administrador por gente no autorizada, debido al extravío de los papeles, donde tienen apuntada la contraseña, que cargan los administradores

A4 Ingeniería Social. Es una amenaza latente con la cual personas ajenas al Objeto de Evaluación podrían hacerse de cuentas para intentar algún ataque

A5 Extravío o perdida de copias de seguridad, debido a que no se encuentran en un lugar fijo y alejado del Objeto de Evaluación

A6 Falla en el *hardware* del servidor: en el único disco duro, en la tarjeta de red, en la memoria RAM, en la tarjeta madre

A7 Falla en la unidad UPS

A8 Robo de equipo, debido a que la jaula en donde se ubica no tiene candado y las paredes con ventanales no tienen una protección metálica

A9 Incendio, debido a una falla en los equipos, lámparas y conectores que rodean al Objeto de Evaluación, aunado a que no existe un sistema contra incendios, ni siquiera un extintor. Además falta un sistema de ventilación

A10 Falta de protección al cableado, no existen canaletas para resguardarlos, además éste cableado está al descubierto en el Área de Seguridad y en las Salas de Cómputo, donde pudiera haber algún accidente, provocado o no, que involucre la rotura del mismo

A11 El acceso a la Bodega donde está el equipo activo, aunque está resguardada, su acceso no es limitado a personal relacionado con las redes y comunicaciones

A12 Si el equipo activo falla, esté hecho puede atentar contra la disponibilidad de muchos equipos incluyendo el Objeto de Evaluación

A13 Interrupción intermitente o prolongada de los servicios de red y electrificación

A14 Descargas de energía eléctrica

A15 Desastre natural

2.4.2 AMENAZAS LÓGICAS

A16 Depende en gran medida del *Firewall* del segmento 132.248.54.0, si este equipo no está disponible, por ende, el Objeto de Evaluación tampoco

A17 Fallas en el *kernel* y en los programas del sistema operativo debido a que no ha sido actualizado. Esto provocaría entre otros problemas la negación de servicio

A18 Modificación, borrado o husmeo en los archivos y directorios tanto del sitio principal ubicados en */casa* así como en los directorios WWW de los usuarios puesto que no tienen los permisos adecuados. (Desfiguramiento de sitios Web)

A19 Robo de cuentas debido a contraseñas inseguras de los usuarios del sistema

A20 Llenado de las particiones */var y /usr* debido a un creciente aumento del tamaño de las bitácoras. Provocaría la negación de servicio

A21 Robo de contraseñas e información debido a que el transporte entre cliente y Servidor Web no está cifrado

A22 Fallas en el *software* Apache, PHP y PostgreSQL debido a las vulnerabilidades existentes ante la falta de actualización. Provocaría la negación de servicio

A23 Subalimentación de los recursos por parte de Apache debido una incorrecta configuración de uso que hace de estos. Por ejemplo si no se limita el uso de memoria, procesos, llamadas, provocaría una negación de servicio

A24 Ejecución arbitraria de CGI's que ya no se utilizan, sin embargo están existentes en el servidor

A25 Fallas en el *software* de seguridad debido a que no ha sido actualizado

A26 Robo de contraseñas o intrusión debido a que algunos sistemas y aplicaciones Web no están restringidos sólo a los equipos que debieran tener acceso

A27 Suplantación de equipos con los cuáles mantiene mecanismos de confianza, debido a que OpenSSH no está actualizado

A28 Robo y mal uso de cuentas que no están siendo ya utilizadas. La falta de información referente a algunas cuentas podría facilitar el anonimato de los atacantes

A29 Ejecución de inyección de código arbitrario en las aplicaciones que estén mal programadas

A30 Llamadas simultáneas al servidor, con la herramienta PING se puede realizar un ataque que consiste en que el atacante envía innumerables llamadas simultáneas al servidor, que se ve saturado al tratar de responder todos los intentos de conexión

2.5 POLÍTICAS DE SEGURIDAD

P1 Para realizar las tareas de administración, operación, mantenimiento y seguridad deberá existir un equipo o *staff* de al menos tres administradores altamente capacitados

P2 Contar con la infraestructura suficiente que permita la emergente reubicación del servidor en caso necesario

P3 Todo administrador deberá memorizar las contraseñas de acceso a la cuenta *root*, evitando su anotación en papel u otro medio que pueda ser visualizado por otra persona

P4 La cuenta de los usuarios es personal e intransferible

P5 Es responsabilidad del usuario el correcto uso de la cuenta. Ante cualquier anomalía observada por el usuario, éste deberá informar a los administradores del suceso

P6 Las copias de seguridad deberán ser etiquetadas, registradas en una bitácora y resguardadas en un lugar alejado de la ubicación de los servidores

P7 Sólo personal autorizado puede manipular las copias de seguridad

P8 El servidor deberá contar con los recursos necesarios para su correcto funcionamiento

P9 Se dará mantenimiento preventivo y correctivo de manera periódica al *hardware* del servidor

P10 Deberán implantarse mecanismos que aseguren la disponibilidad del servidor

P11 Se implantarán sistemas de fuente de energía ininterrumpible

P12 Se dará mantenimiento preventivo y correctivo de manera periódica a los sistemas de fuente de energía ininterrumpible

P13 El servidor estará resguardado en un lugar acondicionado que contará con las protecciones necesarias para su salvaguarda

P14 El acceso al lugar donde se ubica el servidor estará restringido sólo a personal autorizado

P15 Se tendrá una bitácora donde se registrará la persona que entra, anotando nombre, fecha, hora de entrada y salida, así como una breve descripción de la actividad realizada

P16 El lugar donde se aloje el servidor deberá estar acondicionado, contará con sistemas de ventilación, sistemas contra incendios, fuentes suficientes de iluminación, suficientes tomas de corriente

P17 Se dará mantenimiento preventivo y correctivo periódicamente a todos los sistemas de acondicionamiento, contra incendios, iluminación, electrificación

P18 La introducción de alimentos, bebidas y cigarros está restringida

P19 Todo el cableado deberá estar fuertemente protegido, etiquetado y ordenado

P20 Todo el cableado crítico deberá estar replicado

P21 El equipo activo deberá estar resguardado en un lugar acondicionado que contará con las protecciones necesarias para su salvaguarda

P22 El acceso al lugar donde se ubica el equipo activo será restringido y permitido sólo a personal autorizado

P23 Para todo equipo activo crítico, se deberá contar con equipo adicional o redundante que aseguré la disponibilidad de las comunicaciones

P24 Se contará con mecanismos que aseguren la disponibilidad de los servicios de red y electrificación

P25 El servidor contará con reguladores de voltaje y de corriente

P26 El servidor contará con un mecanismo redundante que asegure la disponibilidad del mismo

P27 Contar con la infraestructura suficiente que permita la emergente reubicación del servidor en caso necesario

P28 Proveer los recursos necesarios para garantizar la disponibilidad de todos los dispositivos que comunican al servidor con el exterior

P29 Actualizar y configurar correctamente, una vez que existan actualizaciones disponibles, el sistema operativo del servidor

P30 Es responsabilidad de los usuarios establecer los permisos mínimos necesarios sobre sus archivos

P31 La contraseña de los usuarios deberá ser lo más robusta posible y difícil de adivinar

P32 Es responsabilidad de los usuarios establecer sus contraseñas seguras, cambiarlas periódicamente y hacerlas confidenciales

P33 Cambiar las contraseñas de acceso cada determinado periodo

P34 Evitar que los usuarios permanezcan un tiempo excesivamente prolongado con la misma contraseña

P35 La revisión de las bitácoras será de manera periódica

P36 Establecer mecanismos que garanticen la confidencialidad de la información que viaja a través de Internet

P37 Actualizar, en cuanto exista una actualización disponible, el *software* que emplea el servidor para garantizar la calidad y seguridad de sus servicios

P38 Realizar una correcta configuración del *software* instalado en el servidor de manera que optimice los recursos del mismo

P39 Eliminar todo archivo innecesario del servidor

P40 Actualizar, en cuanto exista una actualización disponible, el *software* que emplea el servidor para garantizar la calidad y calidad de sus servicios

P41 Implantar mecanismos que garanticen el control de acceso a todo sistema informático del servidor

P42 Garantizar la seguridad de los mecanismos de autenticación entre el servidor y todo equipo con el cual mantenga comunicación

P43 Inhabilitar o en su caso eliminar toda cuenta innecesaria

P44 Establecer periodos de gracia para las cuentas de usuario

P45 Toda cuenta deberá estar asignada a un responsable que deberá estar plenamente identificado

P46 Todo sistema informático deberá estar programado con una filosofía orientada a la funcionalidad y a la seguridad

P47 Todo sistema informático implantará mecanismos de validación de entradas de usuario en el servidor y no en el cliente

P48 Implementar mecanismos que filtren los paquetes que son enviamos al servidor

2.6 OBJETIVOS DE SEGURIDAD

O1 Resguardar el Cuarto de Servidores con protecciones físicas y/o lógicas

O2 Resguardar el servidor con protecciones físicas y/o lógicas

O3 Controlar el acceso al Cuarto de Servidores sólo a personal autorizado

O4 Registrar en una bitácora la actividad del Cuarto de Servidores

O5 Acondicionar el Cuarto de Servidores con sistema de ventilación, sistema contra incendios, suficientes fuentes de iluminación y tomas de corriente

O6 Evitar el ingreso de alimentos, bebidas y cigarros al Cuarto de Servidores

O7 Programar el mantenimiento preventivo y correctivo de los sistemas de ventilación, contra incendios, iluminación y tomas de corrientes

O8 Equipar al servidor con una unidad UPS que además de proveer un respaldo de energía, regule el voltaje y la corriente

O9 Programar el mantenimiento preventivo y correctivo de la unidad UPS del servidor

O10 Equipar el servidor con el *hardware* necesario para que realice su función

O11 Programar el mantenimiento preventivo y correctivo del *hardware* del servidor

O12 Diseñar un sistema que garantice la Alta Disponibilidad del servidor

O13 Etiquetar y registrar en una bitácora las copias de seguridad

O14 Resguardar las copias de seguridad en un lugar alejado del Cuarto de Servidores

O15 Ordenar y etiquetar el cableado

O16 Colocar el cableado en canaletas y ductos para su protección

O17 Controlar el acceso a la Bodega, donde se encuentra el equipo activo, sólo a personal autorizado

O18 Acondicionar la Bodega con sistema de ventilación, sistema contra incendios, suficientes fuentes de iluminación y tomas de corriente

O19 Instalar y configurar, para el equipo activo crítico, equipo redundante

O20 Proveer de una unidad UPS al *Firewall* del segmento 132.248.54.0

O21 Implantar mecanismos que garanticen la disponibilidad del *Firewall* del segmento 132.248.54.0

O22 Actualizar el sistema operativo cada vez que una actualización esté disponible

O23 Configurar correctamente el sistema operativo

O24 Ajustar los permisos que tienen los archivos de los usuarios de tal forma que tengan los mínimos y suficientes privilegios

O25 Hacer que las contraseñas de los usuarios sean difíciles de adivinar y lo más robustas y seguras posibles

O26. Implantar mecanismos que garanticen el cambio periódico de las contraseñas de acceso

O27 Inhabilitar o en su caso eliminar toda cuenta innecesaria

O28 Implantar mecanismos que garanticen que las cuentas de usuario no excedan el tiempo de gracia asignado

O29 Asignar un responsable, plenamente identificado, a cada cuenta de usuario

O30 Actualizar el *software* que emplea el servidor cada vez que una actualización esté disponible

O31 Configurar correctamente el *software* del servidor

O32 Implantar mecanismos que controlen el acceso, desde Internet, a los sitios que sólo el personal autorizado puede acceder

O33 Implantar mecanismos que garanticen la confidencialidad del contenido que viaja por Internet

O34 Optimizar la configuración de tal forma que no exista una subalimentación de los recursos asignados

O35 Eliminar todo archivo innecesario del sistema de archivos

O36 Actualizar el *software* de seguridad cada vez que una actualización esté disponible

O37 Configurar correctamente el *software* de seguridad

O38 Implantar de manera correcta, garantizando su funcionamiento, los mecanismos de confianza entre el servidor y los equipos con los cuales mantiene comunicación

O39 Auditar los sistemas y aplicaciones, buscando programas que no realicen una validación de entrada de los usuarios en el servidor, si es el caso, instarlos a que hagan la corrección necesaria

O40 Instar a los administradores que memoricen las contraseñas de acceso y que no las apunten en papeles u otro medio

O41 Planificar la revisión de bitácoras tanto del sistema, del Servidor Web y de la base de datos

O42 Implantar mecanismos que faciliten la revisión de bitácoras

O43 Asignar y capacitar al menos a tres administradores para garantizar que las tareas de administración, operación, mantenimiento y seguridad sobre el servidor siempre se realicen

O44 Informar a los usuarios que es de su responsabilidad toda acción que involucre el uso de su cuenta, además de ser personal e intransferible. Ante cualquier anomalía, que la informen a los administradores del sistema lo antes posible

O45 Diseñar un Plan de Contingencias que contemple la emergente reubicación del servidor en caso necesario

O46 Implantar un sistema robusto de filtrado de paquetes entrantes al servidor

2.7 REQUERIMIENTOS DE SEGURIDAD

Estos Requerimientos consisten de componentes funcionales determinados a partir de *Common Criteria* Versión 2 Parte 2.

Los Requerimientos Funcionales de Seguridad para este Esquema de Seguridad están condensados en la Tabla 2.17:

COMPONENTES FUNCIONALES	DESCRIPCIÓN
FPT_PHP.3	Resistencia a ataque físico
FDP_ACC.1	Subconjunto de control de acceso
FPT_FLS.1	Falla con la preservación de estado seguro
FRU_FLT.2	Tolerancia a fallos limitada
FPT_RCV.2	Recuperación automática
FMT_MTD.1	Administración de TSF data
FDP_IFC.1	Subconjunto de control de flujo de información
FTP_ITC.1	Canal confiable Inter.-TSF
FMT_SMF.1	Especificación de funciones de administración
FIA_ATD.1	Definición de atributos de usuario
FMT_SAE.1	Autorización por tiempo limitado
FMT_SMR.2	Restricción sobre perfiles de seguridad
FPT_ITC.1	Confidencialidad durante la transmisión Inter.-TSF
FRU_RSA.1	Cuotas máximas
FTP_TRP.1	Ruta confiable
FAU_SAA.1	Análisis de violación potencial

Tabla 2.17 Requerimientos de Seguridad

A continuación se describirán dichos requerimientos de seguridad según *Common Criteria*:

FPT_PHP.3 Resistencia a ataque físico

FPT_PHP.3.1 La TSF resistirá [asignación: *escenarios de manipulación física*] a los [asignación: *lista de elementos/dispositivos TSF*] respondiendo automáticamente de tal forma que la TSF no sea violada.

FDP_ACC.1 Subconjunto de control de acceso

FDP_ACC.1.1 La TSF reforzará el [asignación: *control de acceso SFP*] sobre [asignación: *lista de sujetos, objetos, y las operaciones entre sujetos y objetos cubiertos por la SFP*].

FPT_FLS.1 Falla con la preservación de estado seguro

FPT_FLS.1.1 La TSF preservará un estado seguro cuando ocurran los siguientes tipos de fallas:[asignación: *lista de tipos de fallas en la TSF*].

FRU_FLT.2 Tolerancia a fallos limitada

FRU_FLT.2.1 La TSF asegurará la operación de todas las capacidades de la TOE cuando ocurran las siguientes fallas: [asignación: *lista de tipos de fallas*].

FPT_RCV.2 Recuperación automática

FPT_RCV.2.1 Cuando la recuperación automática a partir de [asignación: *lista de fallas/discontinuidades del servicio*] no es posible, la TSF entrará a un modo de mantenimiento dónde la habilidad de regresar a un estado seguro es proporcionada.

FPT_RCV.2.2 Para [asignación: *lista de fallas/discontinuidades del servicio*], la TSF asegurará el regreso de la TOE a un estado seguro usando procedimientos automáticos.

FMT_MTD.1 Administración de TSF data

FMT_MTD.1.1 La TSF limitará la habilidad de [selección: *reinicio, consulta, modificación, borrado, depuración*, [asignación: *otras operaciones*]] las [asignación: *lista de TSF data*] a [asignación: *perfiles autorizados e identificados*].

FDP_IFC.1 Subconjunto de control de flujo de información

FDP_IFC.1.1 La TSF reforzará el [asignación: *control de flujo de información SFP*] sobre [asignación: *lista de sujetos, información, y operaciones cubiertas por la SFP*].

FTP_ITC.1 Canal confiable Inter-TSF

FTP_ITC.1.1 La TSF proporcionará un canal de comunicación entre sí mismo y un producto IT remoto confiable que es lógicamente distinto de otros canales de comunicación y proporcionará identificación segura de sus puntos finales y protección de el canal de datos ante modificación o revelación.

FMT_SMF.1 Especificación de funciones de administración

FMT_SMF.1.1 La TSF deberá ser capaz de interpretar las siguientes funciones administrativas de seguridad: [asignación: *lista de funciones administrativas de seguridad que deben ser proporcionadas por la TSF*].

FIA_ATD.1 Definición de atributos de usuario

FIA_ATD.1.1 La TSF mantendrá la siguiente lista de atributos de seguridad pertenecientes a usuarios individuales: [asignación: *lista de atributos de seguridad*].

FMT_SAE.1 Autorización por tiempo limitado

FMT_SAE.1.1 La TSF restringirá la capacidad, al especificar un tiempo de expiración, para [asignación: *lista de atributos de seguridad para la cual la expiración esta soportada*] a [asignación: *perfiles autorizados e identificados*].

FMT_SAE.1.2 Para cada uno de estos atributos de seguridad, la TSF hará [asignación: *lista de acciones que se llevarán a cabo cada atributo de seguridad*] después de que el tiempo de expiración para el atributo de seguridad indicado haya pasado.

FMT_SMR.2 Restricción sobre los perfiles de seguridad

FMT_SMR.2.1 La TSF mantendrá los roles: [asignación: *perfiles autorizados e identificados*].

FMT_SMR.2.2 La TSF deberá asociar a los usuarios con los perfiles.

FMT_SMR.2.3 La TSF asegurará que las condiciones [asignación: *condiciones para los diferentes perfiles*] estén satisfechos.

FPT_ITC.1 Confidencialidad durante la transmisión Inter-TSF

FPT_ITC.1.1 La TSF protegerá toda TSF data transmitido desde la TSF hacia un producto IT remoto confiable de revelación no autorizada durante la transmisión.

FRU_RSA.1 Cuotas máximas

FRU_RSA.1.1 La TSF reforzará las cuotas máximas de los siguientes recursos: [asignación: *recursos controlados*] que [selección: *usuarios individuales, grupos de usuarios definidos, sujetos*] pueden usar [selección: *simultáneamente, sobre un periodo de tiempo especificado*].

FTP_TRP.1 Ruta confiable

FTP_TRP.1.1 La TSF proporcionará una ruta de comunicación entre sí mismo y usuarios [selección: *remotos, locales*] que lógicamente es distinto de otras rutas de comunicación y proporciona identificación confiable de sus puntos finales y protección de la comunicación ante modificación o revelación.

FTP_TRP.1.2 La TSF permitirá [selección: *la TSF, usuarios locales, usuarios remotos*] iniciar comunicación vía la ruta confiable.

FTP_TRP.1.3 La TSF requerirá el uso de la ruta confiable para [selección: *autenticación inicial del usuario*, [asignación: *otros servicios para los cuales la ruta confiable es requerida*]].

FAU_SAA.1 Análisis de violación potencial

FAU_SAA.1.1 La TSF será capaz de aplicar un conjunto de reglas en el monitoreo de eventos auditables y basados en estas reglas indicar una violación potencial de la TSP.

FAU_SAA.1.2 La TSF forzará las siguientes reglas para el monitoreo de eventos auditables:

a) Acumulación o combinación de [asignación: subconjunto de eventos auditables definido] conocimientos para indicar una violación de seguridad potencial;

b) [asignación: *cualquier otra regla*].

2.8 SUMARIO

Esta sección muestra porqué el Esquema de Seguridad es correcto e internamente consistente. Se demuestra la forma en que fueron determinados tanto los Objetivos como los Requerimientos de Seguridad, y por último se proporcionan las herramientas o acciones a seguir para cubrir los Requerimientos de Seguridad propuestos.

2.8.1 RAZONES DEL PORQUÉ DE LOS OBJETIVOS DE SEGURIDAD DETERMINADOS

Las Figuras 2.11 y 2.12 muestran, respectivamente, el mapeo entre Objetivos de Seguridad y Amenazas así como el mapeo entre Objetivos de Seguridad y Políticas de Seguridad.

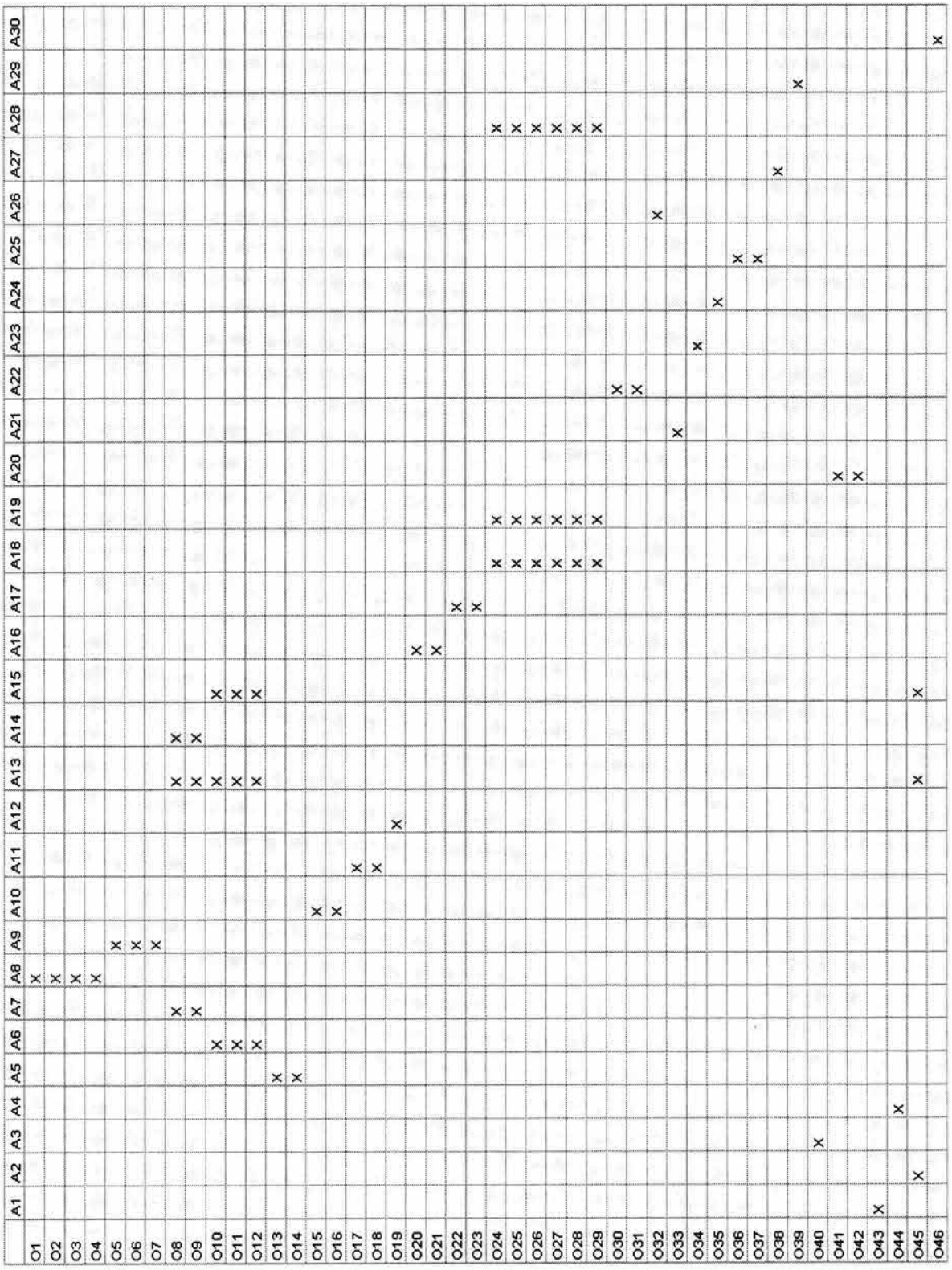

Figura 2.11 Mapeo entre Objetivos de Seguridad y Amenazas

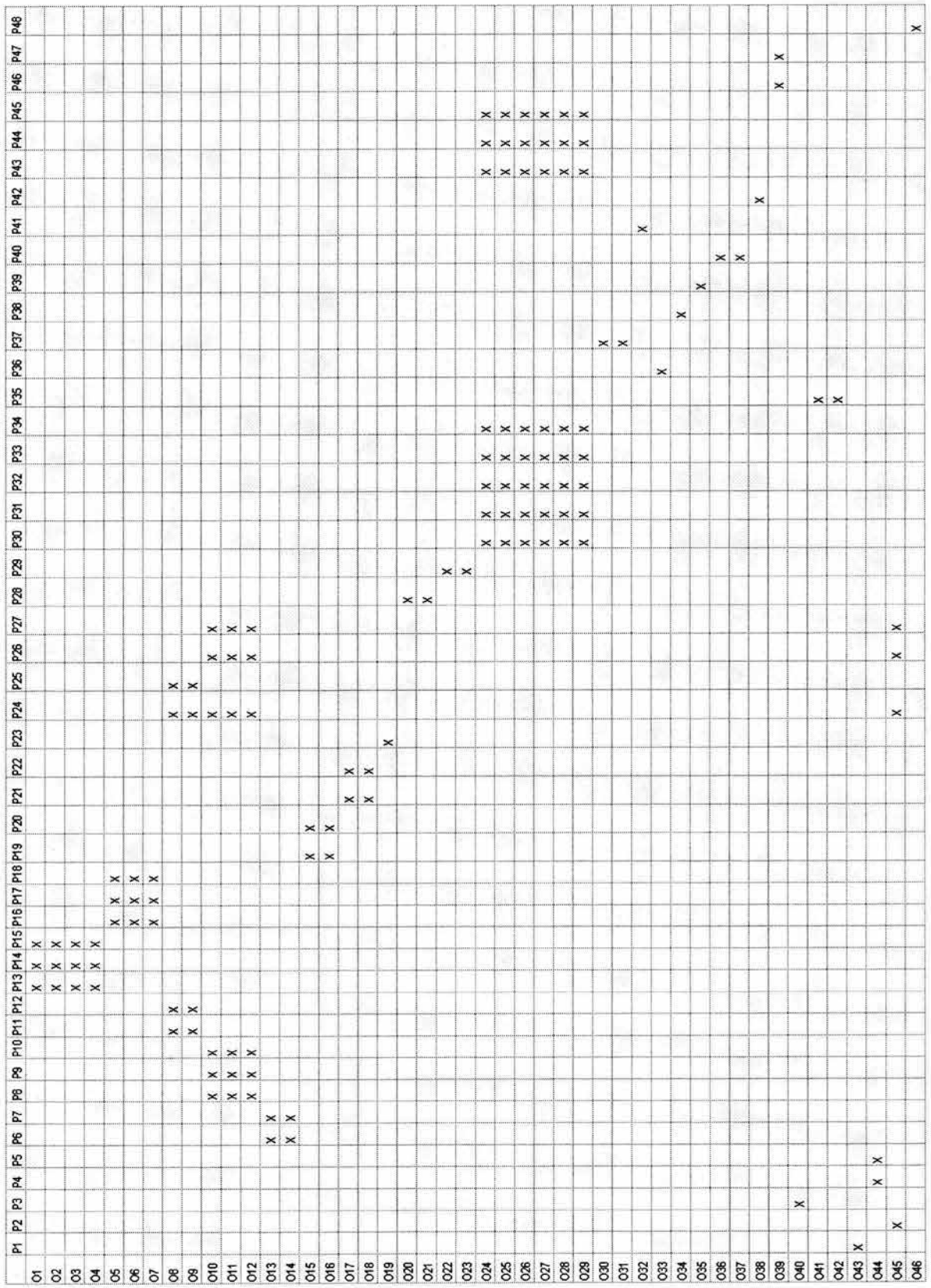

Figura 2.12 Mapeo entre Objetivos de Seguridad y Políticas de Seguridad

Podemos observar que para cada Amenaza y Política de Seguridad existe al menos un Objetivo de Seguridad correspondiéndolo.

O1 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A8 porque se requiere evitar el ataque físico contra el Cuarto de Servidores

O2 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A8 porque se requiere evitar el robo del Objeto de Evaluación

O3 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A8 porque se requiere controlar el acceso al Cuarto de Servidores

O4 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A8 porque se requiere controlar el acceso al Cuarto de Servidores

O5 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A9 porque se requiere proteger al Objeto de Evaluación contra algún siniestro

O6 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A9 porque se requiere proteger al Objeto de Evaluación contra algún siniestro

O7 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A9 porque se requiere proteger al Objeto de Evaluación contra algún siniestro

O8 Este Objetivo de Seguridad es necesario para contrarrestar las amenazas A13 y A14 porque se requiere proteger al Objeto de Evaluación contra el servicio Intermitente, las fluctuaciones y los picos de energía eléctrica

O9 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A7 porque se requiere evitar alguna falla en la unidad UPS

O10 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A6 porque se requiere asegurar la disponibilidad del Objeto de Evaluación

O11 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A6 porque se requiere asegurar la disponibilidad del Objeto de Evaluación

O12 Este Objetivo de Seguridad es necesario para contrarrestar las amenazas A6, A13 y A15 porque se requiere asegurar la disponibilidad del Objeto de Evaluación

O13 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A5 porque se requiere mejorar la administración de las copias de seguridad de la información contenida en el Objeto de Evaluación

O14 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A5 porque se requiere mejorar la administración de las copias de seguridad de la información contenida en el Objeto de Evaluación

O15 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A10 porque se requiere evitar cualquier atentado contra el cableado que conecta al Objeto de Evaluación con el mundo

O16 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A10 porque se requiere evitar cualquier atentado contra el cableado que conecta al Objeto de Evaluación con el mundo

O17 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A11 porque se requiere tener un control de acceso en la Bodega donde se aloja el equipo activo

O18 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A11 porque se requiere salvaguardar la Bodega contra cualquier siniestro

O19 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A12 porque se requiere asegurar la disponibilidad del equipo activo y por ende la del Objeto de Evaluación

O20 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A16 porque se requiere asegurar la disponibilidad del *Firewall* del segmento 132.248.54.0 y por ende la del Objeto de Evaluación

O21 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A16 porque se requiere asegurar la disponibilidad del *Firewall* del segmento 132.248.54.0 y por ende la del Objeto de Evaluación

O22 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A17 porque se requiere evitar la falla del sistema operativo del Objeto de Evaluación

O23 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A17 porque se requiere evitar la falla del sistema operativo del Objeto de Evaluación

O24 Este Objetivo de Seguridad es necesario para contrarrestar las amenazas A18 porque se requiere garantizar la integridad y confidencialidad de la información alojada en el Objeto de Evaluación

O25 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A19 porque se requiere evitar el secuestro de cuentas de usuario debido a contraseñas inseguras

O26 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A19 porque se requiere evitar el secuestro de cuentas de usuario debido a contraseñas inseguras

O27 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A28 porque se requiere evitar el secuestro de cuentas de usuario innecesarias

O28 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A28 porque se requiere evitar el secuestro de cuentas de usuario innecesarias

O29 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A28 porque se requiere plena identificación del responsable de cada cuenta de usuario

O30 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A22 porque se requiere evitar la falla del *software* de aplicación del Objeto de Evaluación

O31 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A22 porque se requiere evitar la falla del *software* de aplicación del Objeto de Evaluación

O32 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A26 porque se requiere controlar el acceso desde Internet a sitios restringidos

O33 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A21 porque se requiere confidencialidad de la información que viaja por Internet entre cliente y Objeto de Evaluación

O34 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A23 porque se requiere controlar el uso de los recursos del Objeto de Evaluación

O35 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A24 porque se requiere optimizar la administración de archivos contenidos en el Objeto de Evaluación

O36 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A25 porque se requiere evitar la falla del *software* de seguridad del Objeto de Evaluación

O37 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A25 porque se requiere evitar la falla del *software* de seguridad del Objeto de Evaluación

O38 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A27 porque se requiere autenticidad entre los equipos de confianza y el Objeto de Evaluación

O39 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A29 porque se requiere identificar y corregir aquellos programas inseguros que atenten contra la seguridad del Objeto de Evaluación

O40 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A3 porque se requiere control de acceso y aseguramiento de la cuenta de administrador

O41 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A20 porque se requiere monitorear la actividad del Objeto de Evaluación

O42 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A20 porque se requiere monitorear la actividad del Objeto de Evaluación

O43 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A1 porque se requiere evitar la ausencia de personal capacitado para atender el Objeto de Evaluación si fuera necesario

O44 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A4 porque se requiere una cultura de seguridad entre los usuarios

O45 Este Objetivo de Seguridad es necesario para contrarrestar las amenazas A2, A13 y A15 porque se requiere asegurar la disponibilidad del Objeto de Evaluación

O46 Este Objetivo de Seguridad es necesario para contrarrestar la amenaza A30 porque se requiere controlar el acceso desde Internet de los paquetes de información que llegan al Objeto de Evaluación

2.8.2 RAZONES DEL PORQUÉ DE LOS REQUERIMIENTOS DE SEGURIDAD DETERMINADOS

En la Figura 2.13 se muestra el mapeo entre los Requerimientos de Seguridad y Objetivos de Seguridad:

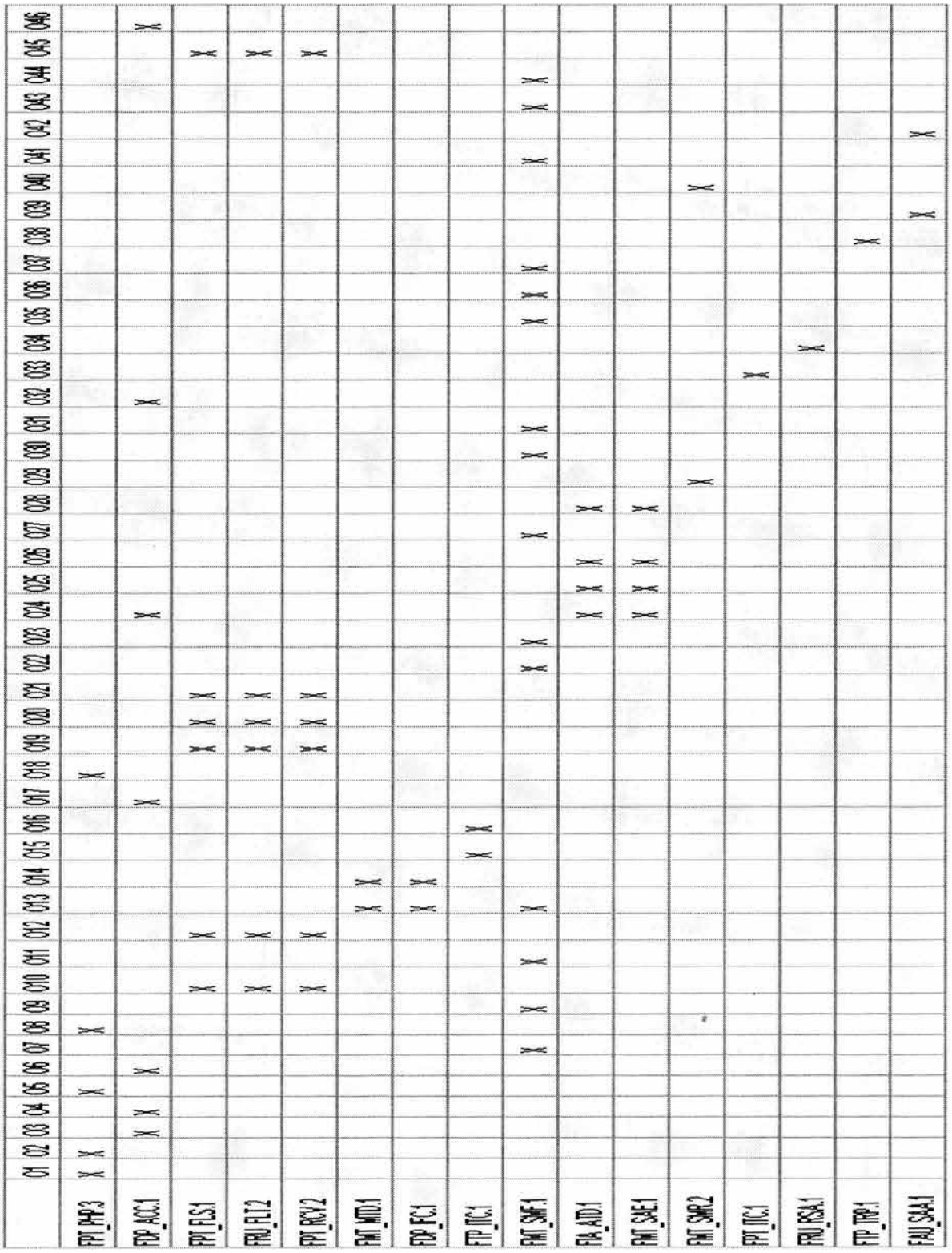

Figura 2.13 Mapeo entre Requerimientos de Seguridad y Objetivos de Seguridad

Podemos observar que para cada Objetivo de Seguridad existe al menos un Requerimiento de Seguridad correspondiéndolo.

FPT_PHP.3 Resistencia a ataque físico

Este Requerimiento de Seguridad garantiza que el Cuarto de Servidores, el Objeto de Evaluación y la Bodega resistirán a ataques físicos

FDP_ACC.1 Subconjunto de control de acceso

Este Requerimiento de Seguridad garantiza el control de acceso al Cuarto de Servidores, a la Bodega, a la información de los usuarios, a los sitios restringidos y al propio Objeto de Evaluación desde Internet.

FPT_FLS.1 Falla con la preservación de estado seguro

Este Requerimiento de Seguridad garantiza el correcto funcionamiento así como la disponibilidad del Objeto de Evaluación, del equipo activo así como del *Firewall* del segmento 132.248.54.0

FRU_FLT.2 Tolerancia a fallos limitada

Este Requerimiento de Seguridad garantiza el correcto funcionamiento así como la disponibilidad del Objeto de Evaluación, del equipo activo así como del *Firewall* del segmento 132.248.54.0

FPT_RCV.2 Recuperación automática

Este Requerimiento de Seguridad garantiza el correcto funcionamiento así como la disponibilidad del Objeto de Evaluación, del equipo activo así como del *Firewall* del segmento 132.248.54.0

FMT_MTD.1 Administración de TSF data

Este Requerimiento de Seguridad garantiza el reforzamiento de la administración de la copias de seguridad de la información alojada en el Objeto de Evaluación

FDP_IFC.1 Subconjunto de control de flujo de información

Este Requerimiento de Seguridad garantiza el reforzamiento de la administración de la copias de seguridad de la información alojada en el Objeto de Evaluación

FTP_ITC.1 Canal confiable Inter-TSF

Este Requerimiento de Seguridad garantiza que el medio sea un canal confiable de comunicación entre el Objeto de Evaluación y el mundo

FMT_SMF.1 Especificación de funciones de administración

Este requerimiento de seguridad garantiza la interpretación de las funciones administrativas de seguridad como son el mantenimiento preventivo y correctivo de los sistemas de acondicionamiento, contra incendio, iluminación, etc.; de las unidades UPS, del *hardware* del Objeto de Evaluación, la administración de copias de seguridad, la administración y manteniendo preventivo y correctivo del sistema operativo de Objeto de Evaluación, del *software* de aplicación y de seguridad, capacitación de personal, entre otros

FIA_ATD.1 Definición de atributos de usuario

Este requerimiento de Seguridad garantiza el mantenimiento de los atributos de seguridad sobre los usuarios como son los permisos sobre los archivos, contraseñas seguras y con fecha de expiración, así como el periodo de gracia para el uso de su cuenta

FMT_SAE.1 Autorización por tiempo limitado

Este Requerimiento de Seguridad garantiza la expiración de contraseñas y cuentas de usuario, entre otros atributos

FMT_SMR.2 Restricción sobre los perfiles de seguridad

Este Requerimiento de Seguridad garantiza la plena identificación de las cuentas de usuario definiendo perfiles de seguridad y condiciones de uso

FPT_ITC.1 Confidencialidad durante la transmisión Inter-TSF

Este Requerimiento de Seguridad garantiza la confidencialidad de la información que viaja a través de Internet entre el cliente y el Objeto de Evaluación

FRU_RSA.1 Cuotas máximas

Este Requerimiento de Seguridad garantiza el correcto uso de los recursos de Objeto de Evaluación

FTP_TRP.1 Ruta confiable

Este Requerimiento de Seguridad garantiza la autenticidad de las comunicaciones entre los equipos de confianza y el Objeto de Evaluación

FAU_SAA.1 Análisis de violación potencial

Este Requerimiento de Seguridad garantiza la capacidad de analizar y detectar cualquier violación potencial contra los sistemas, aplicaciones y contra el mismo sistema operativo Objeto de Evaluación

2.8.3 ACCIONES A SEGUIR

A continuación se muestra un sumario de las herramientas y las acciones a seguir determinadas a partir de los Requerimientos de Seguridad propuestos.

AC1 Reforzamiento, con protecciones metálicas, de los ventanales y puerta de acceso del Cuarto de Servidores

AC2 Colocación de un candado en la jaula metálica donde está el servidor

AC3 Sólo los administradores y el Jefe del Departamento contarán con llaves de acceso

AC4 Crear una bitácora de registro de acceso al Cuarto de Servidores

AC5 Instalar un sistema de ventilación (ventilador), sistema contra incendios (2 extintores)

AC6 Prohibir al personal autorizado la introducción de alimentos, bebidas y cigarros al interior de Cuarto de Servidores

AC7 Realizar una bitácora donde se lleve el control del mantenimiento de los sistemas de ventilación, contra incendios, tomas de corriente, iluminación

AC8 Instalación de una unidad UPS sólo para el servidor

AC9 Realizar una bitácora donde se lleve el control del mantenimiento de la unidad UPS

AC10 Instalación de un disco duro de mayor capacidad, ampliar la memoria RAM

AC11 Realizar una bitácora donde se lleve el control del mantenimiento del servidor

AC12 Realizar un análisis, diseño y desarrollo de un Sistema de Alta Disponibilidad para el servidor

AC13 Crear una plantilla para etiquetar las copias de seguridad, una bitácora donde se lleve el control de las mismas

AC14 Acondicionar un lugar (cubículo del Área de Administración UNIX) para el resguardo de las copias de seguridad

AC15 Rediseñar la topología de la red interna, etiquetando y ordenando el cableado de acuerdo a los estándares de cableado estructurado

AC16 Adquirir e instalar canaletas y ductos para el cableado

AC17 Sólo administradores y el Jefe del Departamento tendrán las llaves de acceso a la Bodega

AC18 Instalar un sistema de ventilación (ventilador), sistema contra incendios (2 extintores)

AC19 Analizar cuál es el equipo activo crítico, e instalar un equipo redundante en caso de falla

AC20 Instalación de una unidad UPS sólo para el *Firewall* del segmento 132.248.54.0

AC21 Realizar un análisis, diseño y desarrollo de un Sistema de Alta Disponibilidad para el *Firewall* del segmento 132.248.54.0

AC22 Verificar continuamente si existe una nueva versión del sistema operativo para su inmediata (planificada) actualización

AC23 Realizar un análisis sobre la configuración del servidor, y en caso de encontrar elementos que puedan comprometer el servidor, realizar la corrección pertinente

AC24 Realizar un análisis sobre los permisos de los archivos de los usuarios y realizar los cambios necesarios con el fin de que tengan los mínimos y suficientes privilegios

AC25 Realizar un análisis sobre la robustez de las contraseñas de los usuarios, en caso de encontrar contraseñas débiles, notificar a los usuarios sobre esto e instarlos a que cambien su contraseña actual por una más segura

AC26 Configurar el sistema de tal forma que, pasado un cierto periodo, notifique a los usuarios sobre su cambio obligatorio de contraseña

AC27 Realizar un análisis sobre las cuentas que son innecesarias, realizar un respaldo e inhabilitarlas o eliminarlas del sistema

AC28 Implantar un mecanismo que establezca un periodo de gracia para que las cuentas de los usuarios caduquen

AC29 Verificar que toda cuenta tenga asignado un responsable plenamente identificado, de no tenerlo, tomar las medidas necesarias para que así sea

AC30 Verificar continuamente si existe una actualización o parche del *software* que emplea el servidor para su inmediata (planificada) actualización

AC31 Realizar un análisis sobre la configuración del *software*, y en caso de encontrar elementos que puedan comprometerlo, realizar la corrección pertinente

AC32 Conocer cuales son los sitios "privados", e implantar el control de acceso a los mismos

AC33 Implantar un medio cifrado (TSL/SSL) en el transporte por el que viajan tanto la solicitud del cliente Web como la respuesta del servidor Web

AC34 Realizar pruebas del rendimiento actual del Servidor Web y con base en los resultados optimizar la asignación de los recursos que emplea

AC35 Conocer cuales programas CGI's no son utilizados, realizar un respaldo y eliminarlos

AC36 Verificar continuamente si existe una actualización o parche del software de seguridad para su inmediata (planificada) actualización

AC37 Realizar un análisis sobre la configuración del *software* de seguridad, y en caso de encontrar elementos que puedan comprometerlo, realizar la corrección pertinente

AC38 Crear las llaves de confianza, con el *software* actualizado, para la implantación de los mecanismos de confianza

AC39 Realizar auditorias sobre los sistemas y aplicaciones con el objetivo de determinar y corregir los errores que atenten contra la funcionalidad y seguridad del sistema

AC40 Instar a los administradores que memoricen las contraseñas de administrador

AC41 Realizar la planeación de las auditorias a las bitácoras del sistema, Servidor Web y servidor de base de datos

AC42 Implantar herramientas que auxilien en la auditoria a las bitácoras, tal como Logcheck y/o Logwatch

AC43 Capacitar al menos a tres administradores con el objetivo de garantizar que las tareas de administración, operación, mantenimiento y seguridad sobre el servidor no se suspendan

AC44 Crear y fomentar una cultura de seguridad en los usuarios del sistema

AC45 Realizar un análisis, diseño y desarrollo de un Plan de Contingencias

AC46 Implantar un sistema de filtrado de paquetes o *Firewall* potente

CAPÍTULO 3

Análisis de la información arrojada por el Esquema de Seguridad en relación a la Disponibilidad

CAPÍTULO 3: ANÁLISIS DE LA INFORMACIÓN ARROJADA POR EL ESQUEMA DE SEGURIDAD EN RELACIÓN A LA DISPONIBILIDAD

3.1 ESTADÍSTICAS

Observemos las siguientes estadísticas, éstas se clasifican en tres rubros: Estadísticas de Uso Diario Mensual, Uso por Hora Mensual y Clientes. El levantamiento de datos se hizo a partir de la fecha de instalación del servidor (febrero de 2004) hasta el mes de junio de 2004.

3.1.1 ESTADÍSTICAS DE USO DIARIO MENSUAL

Las Figuras 3.1, 3.2, 3.3, 3.4 y 3.5 muestran el uso diario que se hace del servidor en los meses de febrero, marzo, abril, mayo y junio respectivamente.

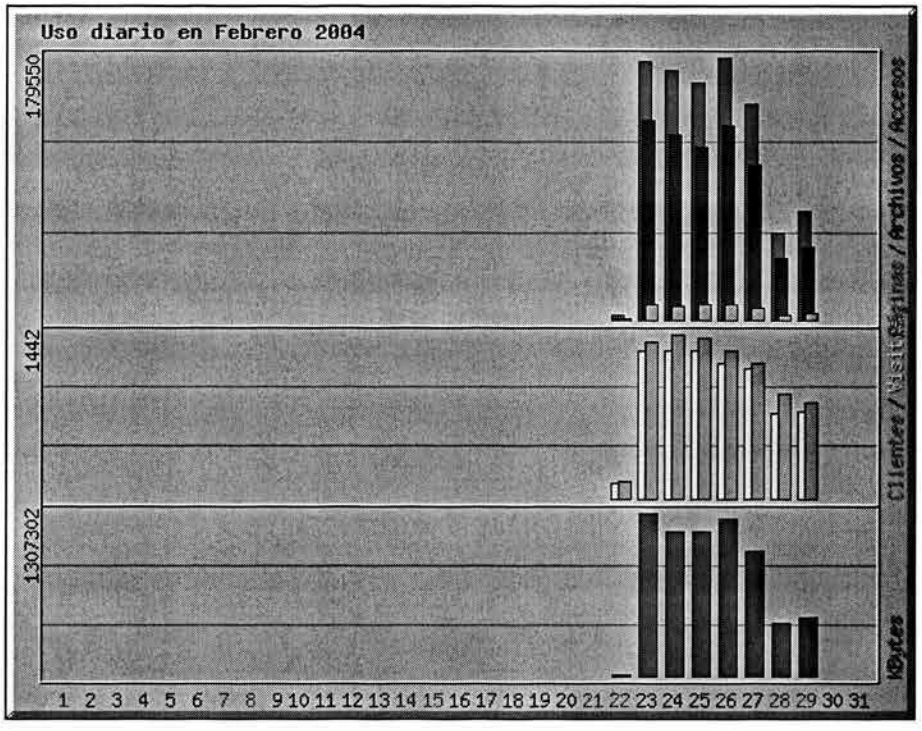

Figura 3.1 Uso Diario en febrero de 2004

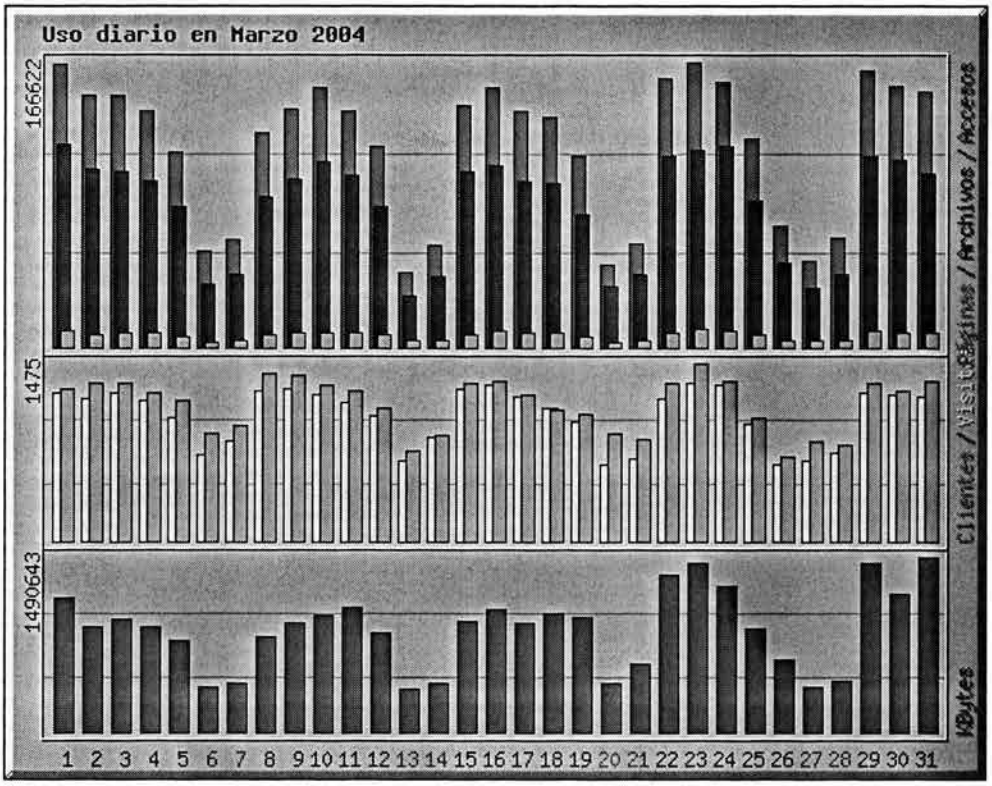

Figura 3.2 Uso Diario en marzo de 2004

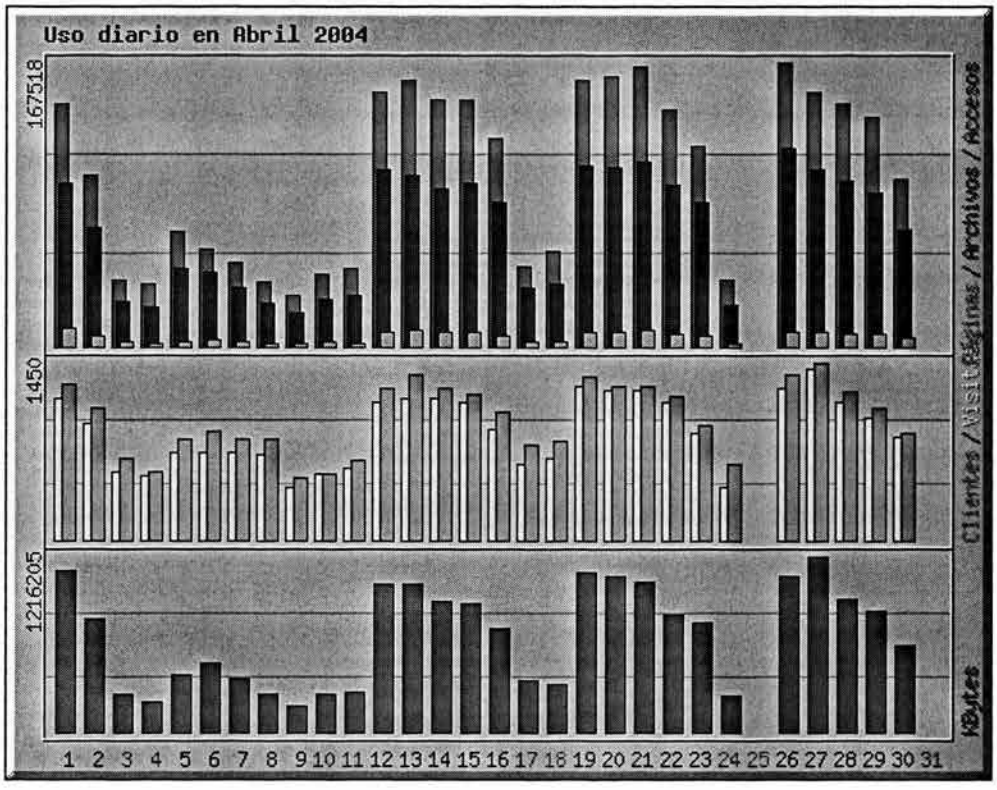

Figura 3.3 Uso Diario en abril de 2004

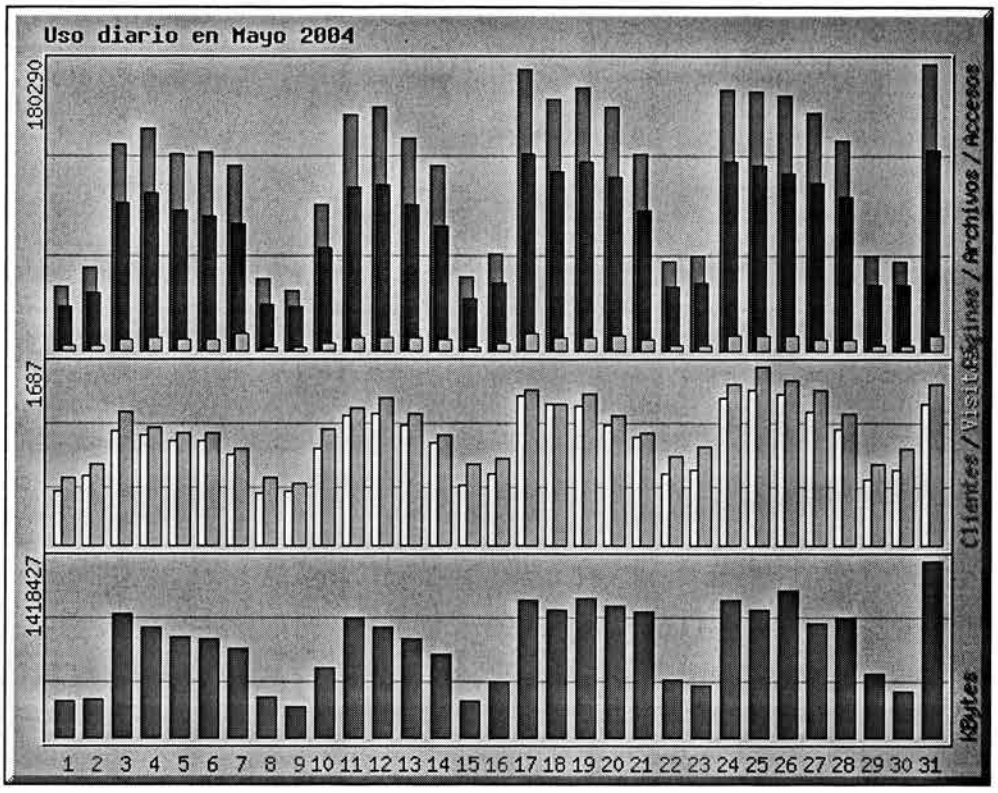

Figura 3.4 Uso Diario en mayo de 2004

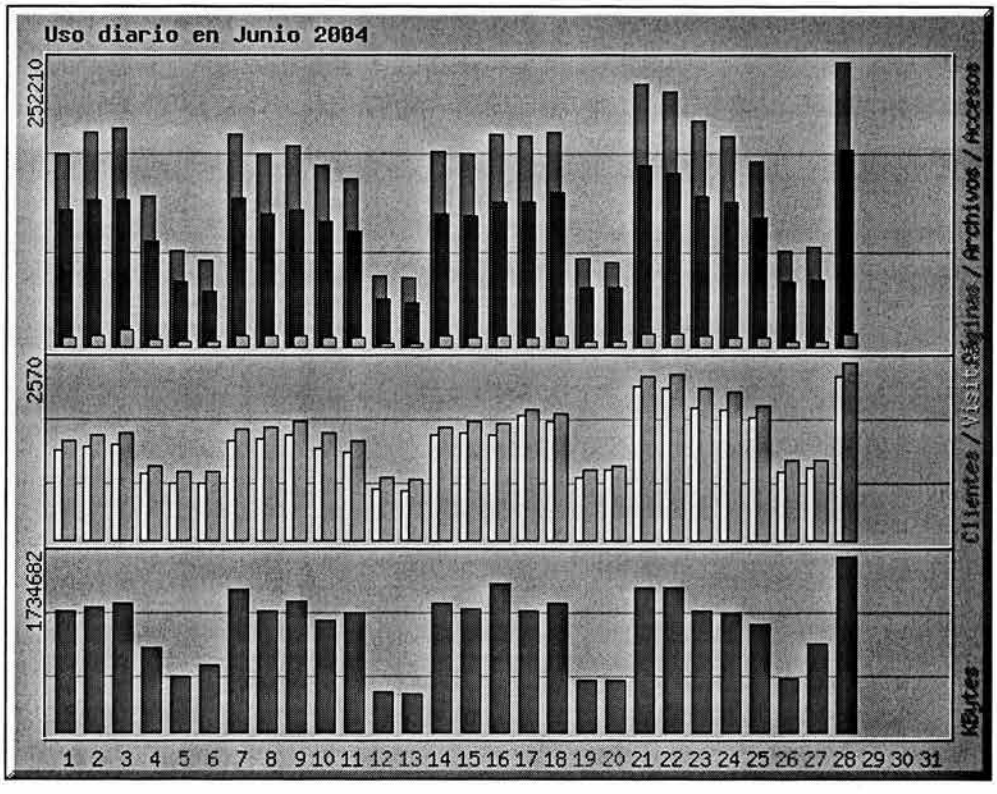

Figura 3.5 Uso Diario en junio de 2004

Podemos observar que durante estos meses, el uso diario del servidor fue en promedio de 189,238 accesos, comparando las gráficas anteriores podemos observar una tendencia clara en cuanto a su uso, por ejemplo, los fines de semana el uso del servidor es menor en comparación al uso que se da entre semana, por tanto su uso es continuo.

3.1.2 ESTADÍSTICAS DE USO POR HORA MENSUAL

Las Figuras 3.6, 3.7, 3.8, 3.9 y 3.10 muestran el uso por hora que se hace del servidor en los meses de febrero, marzo, abril, mayo y junio respectivamente.

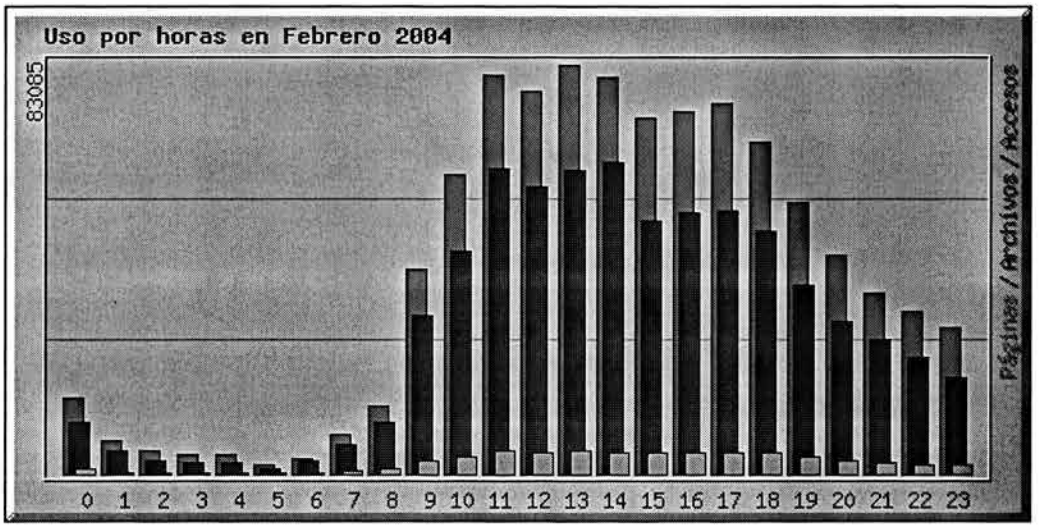

Figura 3.6 Uso por Hora en febrero de 2004

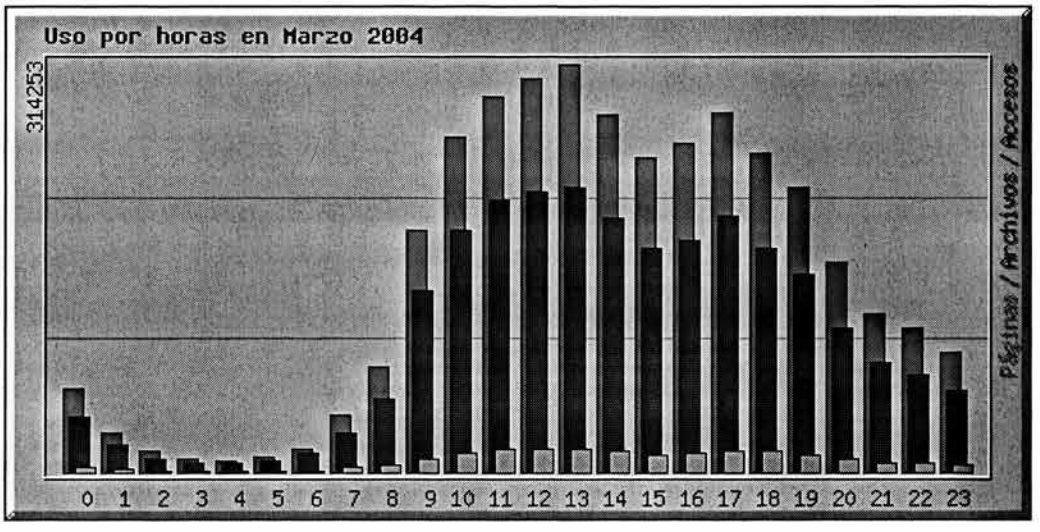

Figura 3.7 Uso por Hora en marzo de 2004

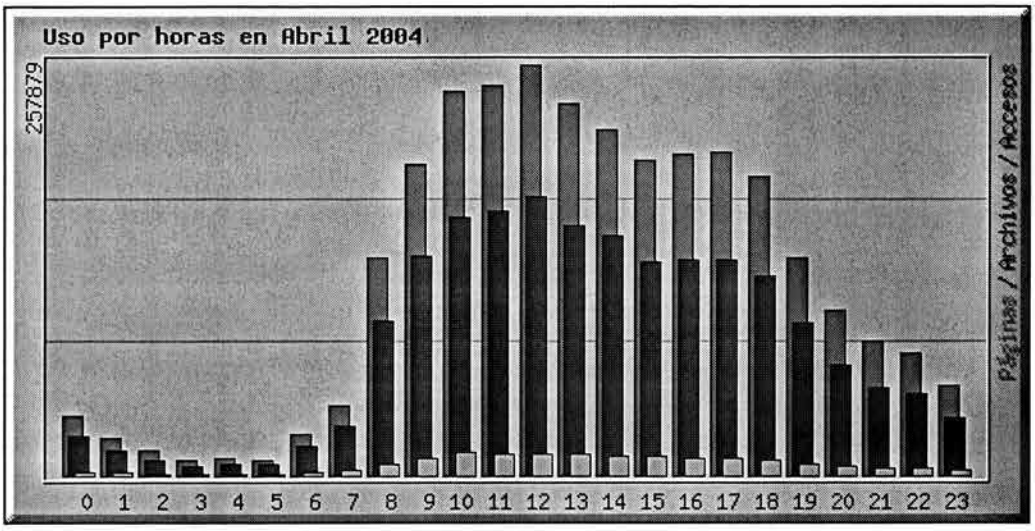

Figura 3.8 Uso por Hora en abril de 2004

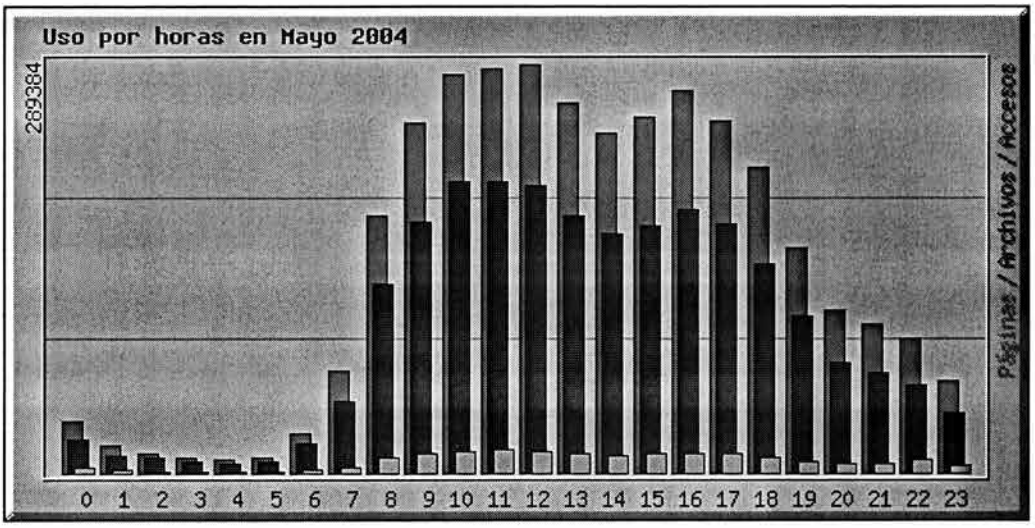

Figura 3.9 Uso por Hora en mayo de 2004

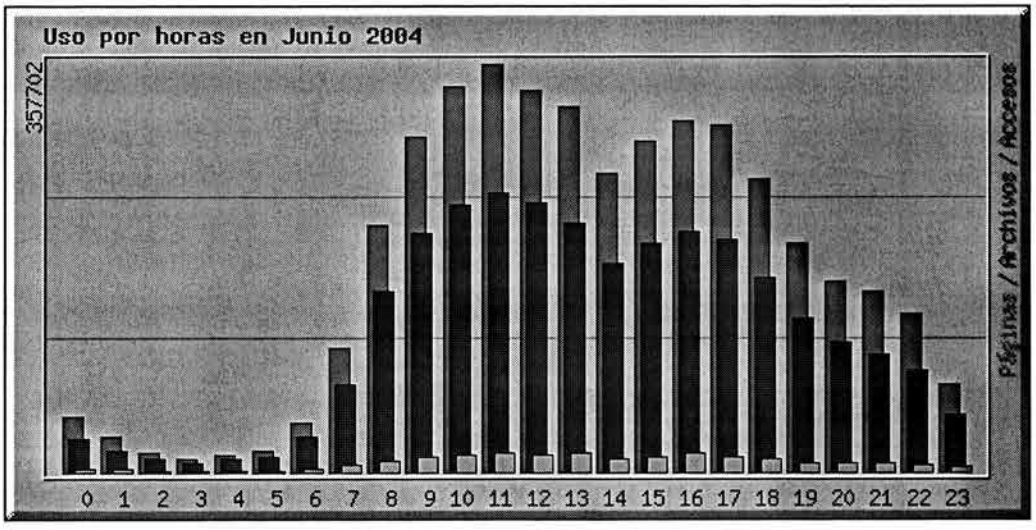

Figura 3.10 Uso por Hora en junio de 2004

En las gráficas anteriores podemos observar una clara tendencia en cuanto al uso por hora del servidor, el periodo crítico de uso sucede a partir de las 8:00 o 9:00 Hrs. hasta antes de las 0:00 Hrs., durante el periodo que va de las 0:00 Hrs. hasta antes de las 8:00 Hrs. el uso decae drásticamente, sin embargo aún existe mínima actividad. Nuevamente, el uso del servidor es continuo.

3.1.3 CLIENTES

La Tabla 3.1 muestra el total de clientes mes a mes durante el periodo de marzo a junio de 2004 que hicieron uso del servidor:

MES	TOTAL DE CLIENTES
Marzo	21,680
Abril	19,442
Mayo	22,990
Junio	29,752

Tabla 3.1 Total de clientes durante el periodo de marzo-junio de 2004

En promedio se tienen 23,466 clientes durante el mes.

Por otro lado las Figuras 3.11, 3.12 y 3.13 muestran el origen de los clientes:

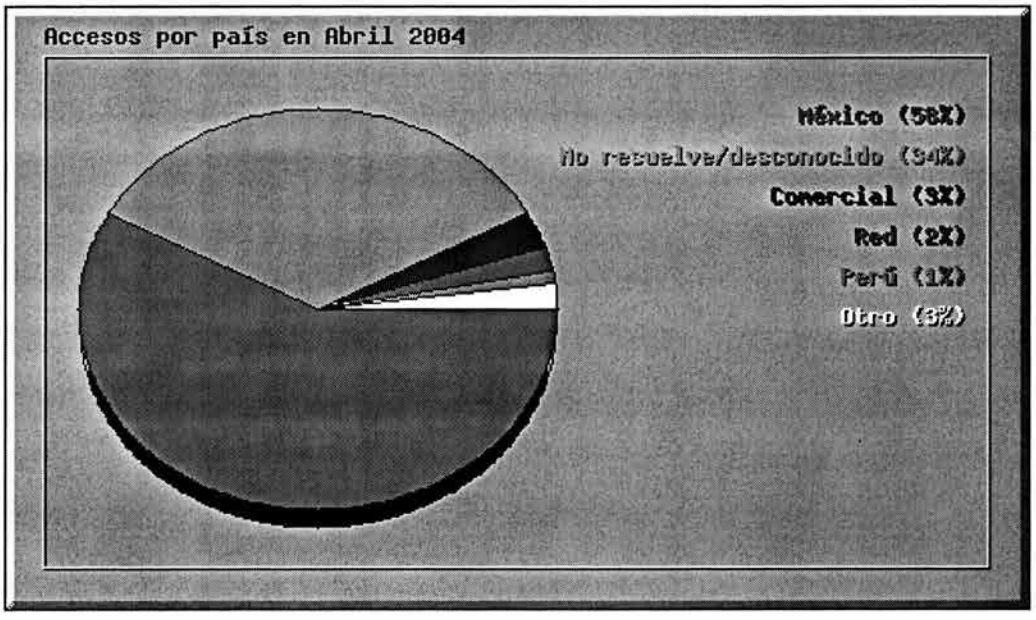

Figura 3.11 Clientes en abril de 2004

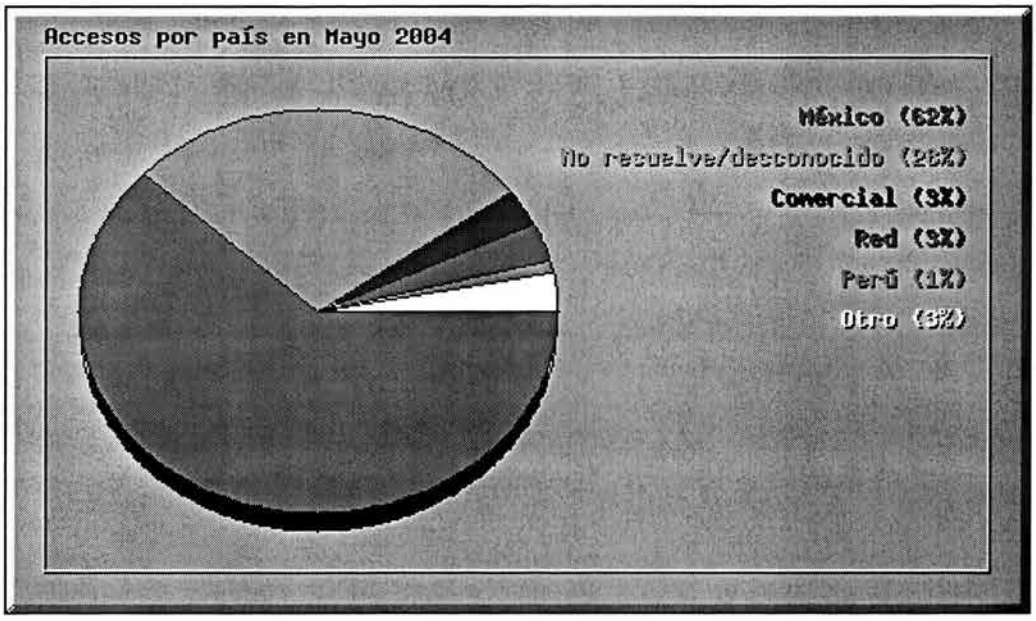

Figura 3.12 Clientes en mayo de 2004

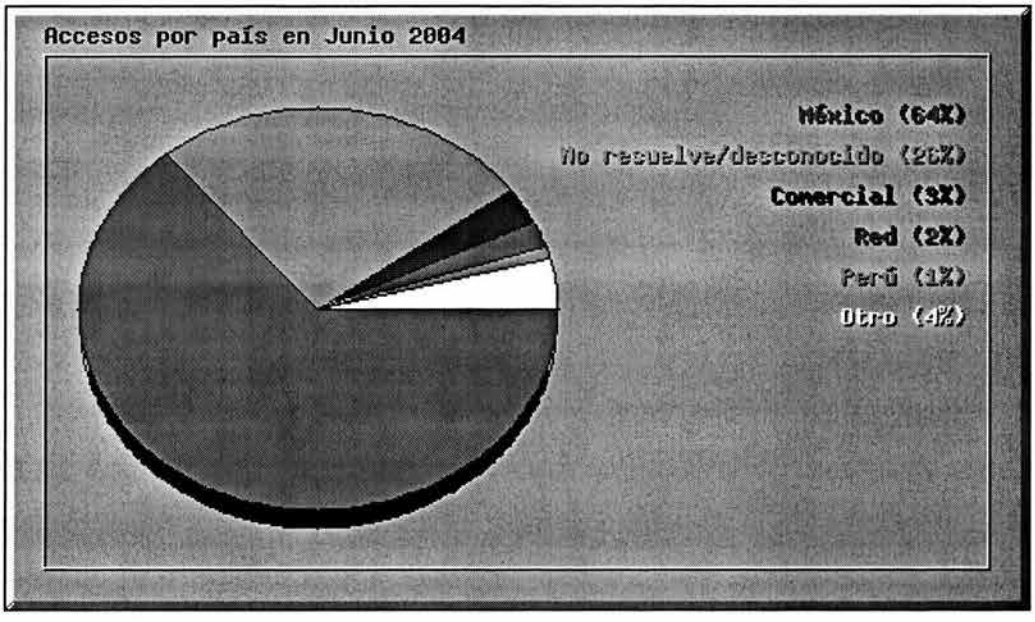

Figura 3.13 Clientes en junio de 2004

A partir de éstas gráficas podemos observar que gran parte de los clientes son originarios de México, en promedio el 61%, sin embargo existen clientes foráneos.

3.2 ¿QUÉ NOS DICE EL ESQUEMA DE SEGURIDAD EN RELACIÓN A LA DISPONIBILIDAD?

El Esquema de Seguridad desarrollado en el Capítulo 2 hace evidente la necesidad de implantar un Sistema de Alta Disponibilidad, hemos visto que la disponibilidad del Objeto de Evaluación es necesaria las 24 horas, los 7 días de la semana, los 265 días del año, incluso ésta es una hipótesis [4]. Existen Puntos Simples de Fallo como son el *hardware* del Objeto de Evaluación, el propio cableado, equipo activo, entre otros que es necesario eliminar. Evidentemente habrá que determinar los alcances del Sistema de Alta Disponibilidad puesto que las amenazas a las que está expuesto van desde la falla de su único disco duro hasta los siniestros e inclusive un desastre natural como un terremoto [5].

Por otro lado existen Políticas de Seguridad que habrá de hacer cumplir y que tienen relación con la disponibilidad de los servicios como son las políticas P8, P10, P26 [6].

Como parte de los Objetivos de Seguridad tenemos a O12 y O45 [7] que ponen en evidencia la necesidad de Alta Disponibilidad de los servicios que ofrece el Objeto de Evaluación.

El Esquema de Seguridad, para el rubro de Disponibilidad, nos da la pauta con la inclusión de los Requerimientos de Seguridad enunciados en la Tabla 3.2 para el diseño de un Sistema de Alta Disponibilidad adecuado:

COMPONENTES FUNCIONALES	DESCRIPCIÓN
FPT_PHP.3	Resistencia a ataque físico
FPT_FLS.1	Falla con la preservación de estado seguro
FRU_FLT.2	Tolerancia a fallos limitada
FPT_RCV.2	Recuperación automática

Tabla 3.2 Requerimientos de Seguridad relacionados con la Alta Disponibilidad [8]

Como parte de las herramientas o acciones a seguir para satisfacer los Requerimientos de Seguridad tenemos a AC12 y AC45 [9] que incluyen el análisis, diseño y desarrollo de mecanismos que proporcionen Alta Disponibilidad además del análisis, diseño y desarrollo de un Plan de Contingencias.

[4] Ver Capítulo 2, sección 2.3
[5] Ver Capítulo 2, sección 2.4
[6] Ver Capítulo 2, sección 2.5
[7] Ver Capítulo 2, sección 2.6
[8] Ver Capítulo 2, sección 2.7
[9] Ver Capítulo 2, sección 2.8.3

CAPÍTULO 4

Diseño del Sistema de Alta Disponibilidad

CAPÍTULO 4: DISEÑO DEL SISTEMA DE ALTA DISPONIBILIDAD

Basándonos en la metodología presentada en el Capítulo 1, desarrollaremos el diseño del Sistema de Alta Disponibilidad para el Servidor Web de la Facultad de Ingeniería.

4.1 DEFINICIÓN DE LAS METAS EN RELACIÓN A LA DISPONIBILIDAD DEL SERVICIO

En esta sección definimos las metas que debe alcanzar el Sistema de Alta Disponibilidad, incluimos una lista detallada de los objetivos del nivel de servicio por medio de un Acuerdo de Nivel de Servicio entre el usuario y el analista-diseñador del Sistema de Alta Disponibilidad.

4.1.1 ACUERDO DE NIVEL DE SERVICIO

Periodo Normal de Operación: El sistema será disponible durante las 24 horas del día, 7 días de la semana y 365 días del año.

Las **Interrupciones Planeadas** son las siguientes (Ver Tabla 4.1):

INTERRUPCIÓN PLANEADA	FRECUENCIA
Reinstalación del servidor	Dos veces por año, cada sesión se lleva 8 horas en promedio
Actualización del sistema operativo y *software* de aplicación	Cada que se presenta una actualización y se emplea alrededor de 2 horas en promedio
Respaldos completos del sistema	Cada dos semanas y se emplea alrededor de 2 horas en promedio
Respaldos incrementales en línea	Cada semana y se emplean 30 minutos en promedio
Suspensión prolongada del servicio de electrificación debido a mantenimiento de la red	Aproximadamente 3 veces al año con duración de 2 o 3 días en promedio

Tabla 4.1 Interrupciones Planeadas

Los **Requerimientos Específicos de Rendimiento** están condensados en la Tabla 4.2:

El tiempo de respuesta a los clientes será de 1 a 2 segundos en condiciones operativas normales, es decir, que no se estén realizando actividades administrativas en línea
El tiempo de respuesta a los clientes durante la realización de respaldos incrementales en línea será de 2 a 3 segundos
El tiempo de recuperación después de sucedida una falla no será mayor a 5 minutos
Todas las consultas o *queries* válidas a la base de datos deben ser satisfechas en un tiempo no mayor a 1 ó 2 segundos
Después de ocurrida una falla el servicio será estable automáticamente
Todas las componentes del servicio estarán en operación después de una falla

Tabla 4.2 Requerimientos Específicos de Rendimiento

Limitaciones:

La suspensión prolongada del servicio de electrificación queda fuera de la jurisdicción o control de la Facultad de Ingeniería y no se cuenta con una fuente de energía alterna como una subestación, por tanto esta interrupción planeada no puede ser contrarrestada puesto que no se considera factible, en este momento, proveer la infraestructura necesaria para colocar un Servidor Web alterno en otra ubicación. Y por este mismo motivo la amenaza de un desastre natural donde esté involucrado el edificio donde se aloja el Servidor Web estará latente y el Sistema de Alta Disponibilidad no responderá adecuadamente, atentando contra la disponibilidad del servicio.

Recursos Necesarios (Ver Tabla 4.3):

Recursos computacionales
Recursos de comunicaciones
Recursos de almacenamiento
Espacio físico alterno
Ambiente físico apropiado
Recursos humanos

Tabla 4.3 Recursos necesarios

Al momento de diseñar o seleccionar la Arquitectura de Alta Disponibilidad que más se adecué a las necesidades planteadas se hará un listado de los recursos tanto computacionales, de comunicaciones así como de almacenamiento que se requerirá, por el momento se hace hincapié en la necesidad de renovar el equipo actual por equipo garantizado y confiable, sumado a que se requiere de repuestos en equipo crítico como son tarjetas de red, discos duros, entre otros.

Por otro lado se requiere de un espacio físico adecuado lo más alejado que se pueda del lugar donde se encuentra el Servidor Web. Esto es para minimizar en lo posible las probabilidades de la falta de disponibilidad debido a un siniestro o desastre donde se vea involucrado el Servidor Web.

Tanto el espacio donde se ubica actualmente el Servidor Web así como el espacio alterno deberán ser acondicionados de manera que provean un ambiente físico apropiado para evitar el sobrecalentamiento, la mala operación de los equipos e inclusive un siniestro.

El recurso humano es parte importante para la migración hacia un Sistema de Alta Disponibilidad, tanto la directiva como los administradores y operadores deberán estar concientes y pensar, en lo relacionado al sistema, en términos de Alta Disponibilidad. La participación, la clara definición de acciones y funciones y la disciplina facilitarán la implantación. Es de vital importancia que el recurso humano tome conciencia del papel que desempeña como elemento activo del Sistema de Alta Disponibilidad.

4.2 DETERMINACIÓN DE LA DURACIÓN MÁXIMA DE UNA INTERRUPCIÓN ACEPTABLE

Según los datos expuestos en el Acuerdo de Nivel de Servicio, la duración máxima de una interrupción aceptada por el usuario será de 5 minutos.

4.3 DETERMINACIÓN DE LA DISPONIBILIDAD DEL SISTEMA

4.3.1 PORCENTAJE DE DISPONIBILIDAD

Se requiere que el sistema esté disponible las 24 horas del día, los 7 días de la semana y los 365 días del año, se piensa que habrá aproximadamente durante el año 8 interrupciones planeadas o no planeadas del servicio. Y para restablecer el servicio se contemplan 5 minutos aproximadamente.

Con estos datos, podemos calcular la disponibilidad del sistema de acuerdo con las Fórmulas 1.6 y 1.7 del Capítulo 1, Sección 1.3.1:

$$MTBF = \frac{8760 HORAS}{8 INTERRUPCIONES} = 1095 HORAS$$

Y $$MTTR = 0.083 HORAS$$

Por lo tanto:

$$DISPONIBILIDAD = \frac{1095 HORAS}{1095 HORAS + 0.083 HORAS} = 0.999924206658308091715422483957 84$$

En porcentaje y aplicando un redondeo a dos cifras decimales:

$$\%DISPONIBILIDAD = 99.99$$

Con este dato, podemos decir que para el sistema 24X7X365 de acuerdo a los Porcentajes de Disponibilidad comerciales tenemos lo siguiente (Ver Tabla 4.4):

%UPTIME	99.99
%DOWNTIME	0.01
TIEMPO SIN SERVICIO ANUAL	52 minutos, 30 segundos
NORMALIZADO EN SEGUNDOS	3,150

Tabla 4.4 Perfil de Disponibilidad requerido

4.3.2 NIVEL DE DISPONIBILIDAD

En función del Porcentaje de Disponibilidad y de los requerimientos del usuario explícitamente expresados en el Acuerdo de Nivel de Servicio podemos determinar el Nivel de Disponibilidad adecuado para el sistema (Ver Tabla 4.5).

NIVEL	Tolerancia a Fallos
FUNCIONES	Capacidad de procesamiento continuo y cualquier fallo debe ser transparente para el usuario
PORCENTAJE DE DISPONIBILIDAD	Menor al 99.99%
SISTEMA DE ALTA DISPONIBILIDAD SUGERIDO	Sistema Tolerante a Fallos: Van un paso más lejos para asegurarse de que no habrán interrupciones en los sistemas. Para alcanzar este nivel de disponibilidad, se deben utilizar servidores duales o los *clusters* de servidores redundantes donde un servidor asume el control automáticamente si el otro servidor cae
MECANISMOS BÁSICOS A IMPLANTAR	Duplicidad y redundancia

Tabla 4.5 Nivel de Disponibilidad requerido

4.4 DEFINICIÓN DEL GRUPO DE SERVICIOS Y GRUPO DE RECURSOS

El Grupo de Servicios está constituido por (Ver Tabla 4.6):

Servicio Web
Servicio de base de datos

Tabla 4.6 Grupo de Servicios

El Grupo de Recursos está constituido por (Ver Tabla 4.7):

Red de comunicaciones
Firewall del segmento 132.248.54.0
SPU
Unidad de almacenamiento
Fuente de poder
Software
Ser humano

Tabla 4.7 Grupo de Recursos

4.5 IDENTIFICACIÓN DE LOS PUNTOS SIMPLES DE FALLO EN EL SISTEMA ACTUAL

Con base en el Grupo de Recursos definido podemos identificar los siguientes Puntos Simples de Fallo (Ver Tabla 4.8):

GRUPO DE RECURSO	PUNTO SIMPLE DE FALLO
Red de comunicaciones	Equipo activo Cableado que va de éste hacia el Servidor Web Única tarjeta de red
Firewall del segmento 132.248.54.0	Si éste equipo falla la disponibilidad del servicio es nula
SPU	Memoria RAM Tarjeta Madre Tarjetas de expansión Microprocesador Ventilación Disipador
Unidad de almacenamiento	Único disco duro
Software	Sistema operativo Servidor Web Manejador de base de datos
Ser humano	Procedimientos ante falla o contingencia no definidos

Tabla 4.8 Puntos Simples de Fallo identificados

El Grupo de Recursos "fuente de poder" no se identifica como Punto Simple de Fallo puesto que en el esquema actual, el Servidor Web cuenta con una unidad UPS.

4.6 EVALUACIÓN DE LAS APLICACIONES

Sobre las aplicaciones como son el Servidor Web y el manejador de base de datos, ambos cumplen con las siguientes características (Ver Tabla 4.9):

Son configurables sin necesidad de reiniciar el sistema, sólo se reinicia la aplicación
Cuentan con herramientas de administración
Se recuperan después de un fallo
Pueden ser ejecutadas en más de un nodo

Tabla 4.9 Características de las aplicaciones

Por tanto, al contar con estas características, la tarea de proveer Alta Disponibilidad a nivel de aplicación sale favorecida.

4.7 DISEÑO O SELECCIÓN DE LA ARQUITECTURA DE ALTA DISPONIBILIDAD

De acuerdo al Nivel de Disponibilidad determinado, el Sistema de Alta Disponibilidad sugerido es el Sistema Tolerante a Fallos, que básicamente consiste en un *cluster* de servidores redundantes y los mecanismos a utilizar, para asegurar el Nivel y Porcentaje de Disponibilidad requerido, serán la duplicidad y redundancia.

4.7.1 ARQUITECTURA SELECCIONADA

En el rubro de *clusters*, podemos seleccionar entre tres arquitecturas: *Activa-Pasiva, Activa-Activa y Base de datos paralela.*

La arquitectura que considero más adecuada para el Sistema de Alta Disponibilidad es la Activa-Activa de dos nodos, porque se requiere que el nodo alterno no sólo esté esperando la falla del nodo principal, sino que esté "activamente" participando como respaldo de información además de ejecutar aplicaciones de misión crítica que en este caso son las aplicaciones Web y de base de datos.

¿Por qué sólo dos nodos? Considero que con un nodo de respaldo es suficiente para mantener continuo el servicio, mientras que el nodo que sufrió fallo es atendido y puesto en marcha. La ventaja se ve reflejada en la asignación de recursos, puesto que los recursos para un *cluster* de tres o más nodos pueden ser aprovechados en otros aspectos del Sistema de Alta Disponibilidad. La desventaja es que una vez que un nodo falle, entonces el nodo que toma su lugar se convierte en un Punto Simple de Fallo.

4.7.2 REQUERIMIENTOS PARA LA CREACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD

En la Tabla 4.10 se definen los principales requerimientos que deberá cumplir el *cluster* de Alta Disponibilidad para que sea adecuado a las necesidades planteadas:

La arquitectura del *cluster* será Activa-Activa
En ambos servidores se ejecutarán aplicaciones de misión crítica
El nodo alterno servirá como respaldo
Se implantarán mecanismos que aseguren la replicación cercana al 100% de los datos en el nodo alterno
El tiempo en el cual un nodo suplanta al otro no deberá ser mayor a 5 minutos
La dinámica del *cluster* deberá incluir: *failover, takeover, switchover y splitbrain*
El *cluster* deberá implantar un mecanismo que permita notificar inmediatamente de la caída de un nodo para su pronta recuperación
El *cluster* mantendrá una bitácora sobre su actividad
Registrará los problemas que atiende así como las acciones correctivas aplicadas
Registrará las interrupciones planeadas y no planeadas
El *cluster* incorporará una interfaz de administración

Tabla 4.10 Requerimientos que deberá cubrir el Cluster de Alta Disponibilidad

4.7.3 SUMARIO DE RECURSOS NECESARIOS

Recursos computacionales	• Dos equipos: uno para configurar un *Firewall* redundante y otro para el *Cluster* de Alta Disponibilidad del Servidor Web
Recursos de comunicaciones	• Equipo activo, al menos dos *switches* • Cable UTP Categoría 6 • 6 tarjetas de red • Canaletas y ductos
Recursos de almacenamiento	• Al menos un disco duro de la misma capacidad que el que tiene actualmente el Servidor Web
Espacio alterno	• Un cubículo lo más alejado posible del Cuarto de Servidores
Ambiente físico apropiado	• Protecciones metálicas para los ventanales del Cuarto de Servidores • Un candado para la jaula • Dos ventiladores • Cuatro extintores • Papelería para bitácoras
Recursos humanos	• Para la operación y mantenimiento del Sistema de Alta Disponibilidad

Tabla 4.11 Recursos necesarios

4.7.4 LISTA DE ACTIVIDADES A DESARROLLAR

En el siguiente sumario (Ver Tabla 4.12) se resumen las actividades a desarrollar para la creación del Sistema de Alta Disponibilidad.

Construcción del ambiente físico apropiado
Creación de procesos automatizados
Eliminación de: la red, *Firewall* del segmento 132.248.54.0, unidad de almacenamiento, *software* y ser humano como Puntos Simples de Fallo
Eliminación de la SPU como Punto Simple de Fallo: Creación del *Cluster* de Alta Disponibilidad

Tabla 4.12 Sumario de actividades

CAPÍTULO 5

Desarrollo del Sistema de Alta Disponibilidad

CAPÍTULO 5: DESARROLLO DEL SISTEMA DE ALTA DISPONIBILIDAD

Una vez realizado el diseño del Sistema de Alta Disponibilidad adecuado para el Objeto de Evaluación, la siguiente etapa del proceso es el desarrollo del propio sistema. En general, este desarrollo consiste en la construcción de un ambiente físico apropiado para el sistema, la creación de procesos automatizados, eliminación de los Puntos Simples de Fallo determinados en el diseño y por último la creación del *Cluster* de Alta Disponibilidad como elemento que elimina a la SPU como Punto Simple de Fallo.

5.1 CONTRUCCIÓN DEL AMBIENTE FÍSICO APROPIADO

Para la creación de un ambiente físico apropiado, nos apoyaremos en el Esquema de Seguridad desarrollado en el Capítulo 2, documento que determinó tanto las Políticas y los Requerimientos de Seguridad necesarios para garantizar la protección física del Objeto de Evaluación. [10] El cumplimiento de las Políticas P13, P14, P15, P16, P17, P18, P19, P21, P22 y P25, además de la plena satisfacción de los Requerimientos FPT_PHP.3, FPD_ACC.1, FMT_SMF.1 y FTP_ITC.1 garantizan un ambiente físico apropiado.

De acuerdo al Esquema de Seguridad las herramientas aplicables y las acciones a seguir para crear un ambiente físico apropiado son las siguientes:

AC1 Reforzamiento, con protecciones metálicas, de los ventanales y puerta de acceso del Cuarto de Servidores

AC2 Colocación de un candado en la jaula metálica donde está el servidor

AC3 Sólo los administradores y el Jefe del Departamento contarán con llaves de acceso

AC4 Crear una bitácora de registro de acceso al Cuarto de Servidores

AC5 Instalar un sistema de ventilación (ventilador), sistema contra incendios (2 extintores)

AC6 Prohibir al personal autorizado la introducción de alimentos, bebidas y cigarros al interior de Cuarto de Servidores

AC7 Realizar una bitácora donde se lleve el control del mantenimiento de los sistemas de ventilación, contra incendios, tomas de corriente, iluminación

[10] Ver Capítulo 2, secciones 2.5, 2.7 y 2.8.3

AC8 Instalación de una unidad UPS sólo para el servidor

AC9 Realizar una bitácora donde se lleve el control del mantenimiento de la unidad UPS

AC15 Rediseñar la topología de la red interna, etiquetando y ordenando el cableado de acuerdo a los estándares de cableado estructurado

AC16 Adquirir e instalar canaletas y ductos para el cableado

AC17 Sólo administradores y el Jefe del Departamento tendrán las llaves de acceso a la Bodega

AC18 Instalar un sistema de ventilación (ventilador), sistema contra incendios (2 extintores)

5.2 CREACIÓN DE PROCESOS AUTOMATIZADOS

Los procesos que son enteramente candidatos para ser automatizados son los respaldos incrementales en línea y la recuperación del sistema ante un fallo.

Ambos procesos serán automatizados por el Sistema de Alta Disponibilidad, en particular el *Cluster*, dentro de sus actividades, tiene contemplado el proceso de respaldo continuo e incremental del Servidor Web, así como la recuperación automática ante un fallo.

5.3 ELIMINACIÓN DE LA RED, EL FIREWALL, LA UNIDAD DE ALMACENAMIENTO, EL SOFTWARE Y EL SER HUMANO COMO PUNTOS SIMPLES DE FALLO

5.3.1 ELIMINACIÓN DE LA RED COMO PUNTO SIMPLE DE FALLO

En el Diseño del Sistema de Alta Disponibilidad se identificó la red de comunicaciones como Punto Simple de Fallo, por ello habrá que eliminarlo. De acuerdo a los datos arrojados por el Esquema de Seguridad en cuanto a la configuración de la red de comunicaciones, Internet y el Objeto de Evaluación se comunican de acuerdo a lo mostrado en la Figura 5.1:

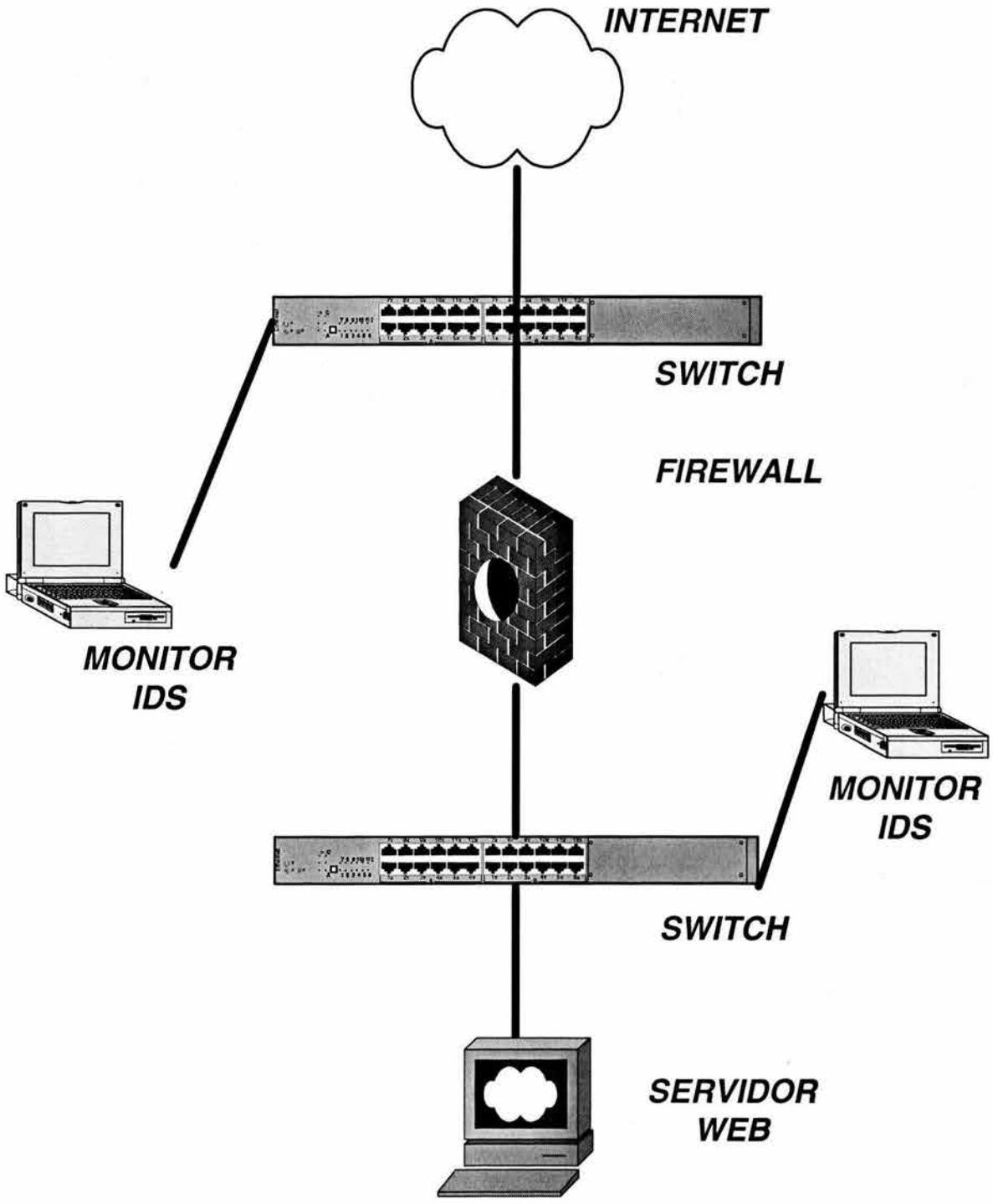

Figura 5.1 Configuración actual de la red de comunicaciones

Los Puntos Simples de Fallo identificados son los siguientes (Ver Tabla 5.1):

Cableado de red
Switches
Tarjetas de red
Firewall

Tabla 5.1 Puntos Simples de Fallo identificados

Podemos eliminar estos Puntos Simples de Fallo proporcionando conexiones LAN redundantes en la medida de lo posible.

Por tanto el Sistema de Alta Disponibilidad requiere una nueva configuración sin Puntos Simples de Fallo como se ilustra en la Figura 5.2:

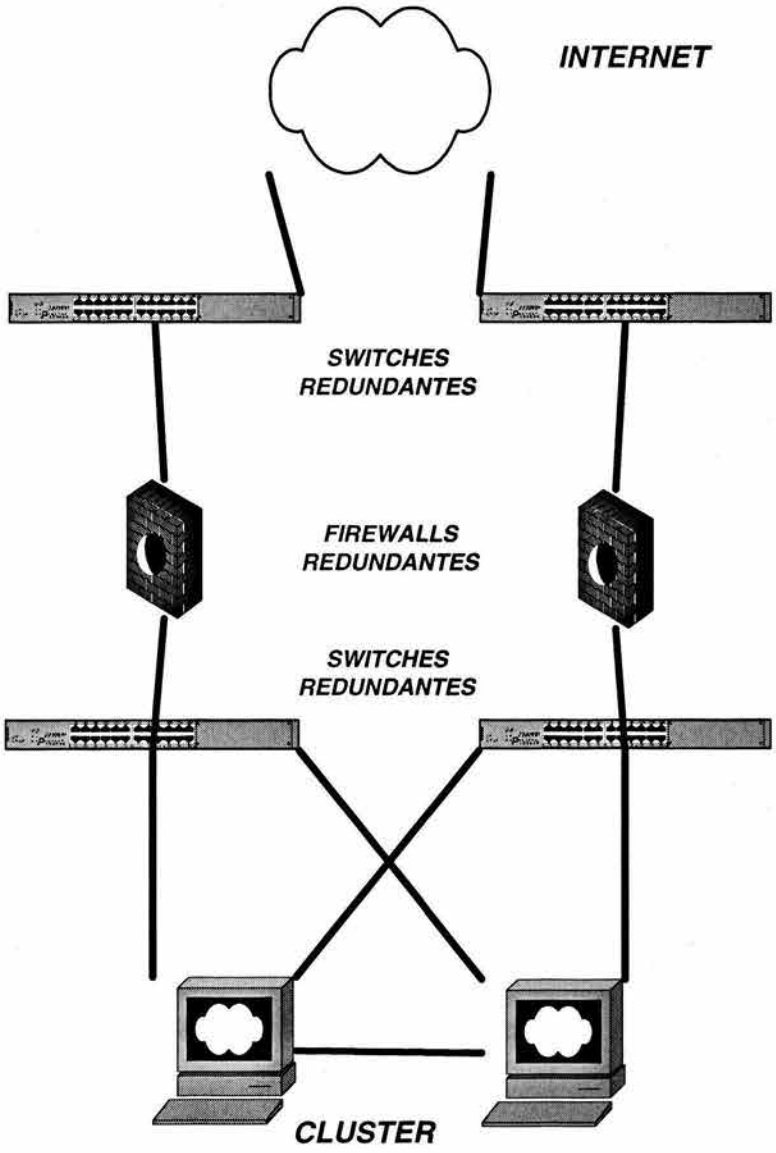

Figura 5.2 Alta Disponibilidad en la red de comunicaciones

Podemos configurar un *switch* redundante para contrarrestar el fallo de otro, en cuanto al *Firewall*, puesto que está implementado a nivel *software*, podemos configurar otro para su replicación. Las tarjetas de red están configuradas de tal forma que una esté en un estado pasivo a la espera del fallo de la tarjeta activa. El cableado también es redundante.

5.3.2 ELIMINACIÓN DEL FIREWALL COMO PUNTO SIMPLE DE FALLO

El *Firewall* es un dispositivo muy importante en lo que respecta al control de acceso hacia el interior de la red de telecomunicaciones. Como lo ilustra la Figura 5.1, actualmente el *Firewall* es un Punto Simple de Fallo. Para eliminar este Punto Simple de Fallo podemos configurar un *Cluster* de Alta Disponibilidad para garantizar su disponibilidad, puesto que sin este dispositivo al menos el Objeto de Evaluación no estaría disponible aún cuando esté funcionando correctamente, perderíamos conectividad con el exterior.

Otro aspecto importante es el de proveer una unidad UPS exclusiva al *Firewall*, si es que no cuenta con ella, para contrarrestar en lo posible la falta de disponibilidad ante la falta intermitente de energía.

5.3.3 ELIMINACIÓN DE LA UNIDAD DE ALMACENAMIENTO COMO PUNTO SIMPLE DE FALLO

El Objeto de Evaluación actualmente cuenta con un solo disco de almacenamiento secundario, esto lo convierte en un Punto Simple de Fallo. Nuevamente la redundancia es necesaria para prevenir que el fallo del disco provoque una interrupción del servicio.

Para ello emplearemos la técnica llamada *Disk Mirroring*, a través de la configuración RAID Nivel 1. A grandes rasgos con este nivel creamos un único disco lógico usando para ello dos discos físicos, todos los datos que se almacenan en el disco lógico son almacenados en ambos discos físicos de forma que son gemelos en todo momento.

En particular aplicaremos una especialidad en Nivel 1 llamada *Drive Duplexing*, que consiste en usar dos controladoras de disco independientes, esto nos proporcionará niveles de tolerancia a fallos máximos.

La configuración RAID Nivel 1 la implantaremos al momento de instalar el sistema operativo, en el momento de crear las particiones necesarias. Para ello requeriremos de otro disco duro y otra controladora de disco.

5.3.4 ELIMINACIÓN DEL SOFTWARE COMO PUNTO SIMPLE DE FALLO

El *software* que emplea el Objeto de Evaluación es el Sistema Operativo, el Servidor Web y el Manejador de Base de Datos. Los productos que se emplean son *Linux Fedora Core, Apache Web Server y PostgreSQL*, respectivamente.

Estos productos son parte de la corriente del *Software Libre*, altamente confiables y, como lo determinamos al evaluar las aplicaciones en el diseño del sistema, cuentan con las características deseables en un entorno de Alta Disponibilidad.

Por tanto, debido a su propia naturaleza, el *software* no es considerado un Punto Simple de Fallo. [11]

5.3.5 ELIMINACIÓN DEL SER HUMANO COMO PUNTO SIMPLE DE FALLO

Se considera al ser humano (administradores y operadores de red) como Punto Simple de Fallo puesto que con sus acciones u omisiones no acertadas pueden provocar la pérdida del servicio hasta que el error humano sea corregido.

Para eliminar este Punto Simple de Fallo, se deben automatizar, en la medida de lo posible, las operaciones, además de documentar cada uno de los procedimientos. Para ello en la sección 5.2 de este capítulo, hemos determinado cuales son los procesos a automatizar. Sin embargo no todas las operaciones están incluidas, tales como la instalación, actualización del sistema y *software* y los respaldos completos del sistema. Para estas actividades documentaremos los procedimientos en el capítulo siguiente relacionado a la implantación del Sistema de Alta Disponibilidad.

5.4 ELIMINACIÓN DEL SPU COMO PUNTO SIMPLE DE FALLO: CREACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD

Esta etapa de creación del *Cluster* de Alta Disponibilidad se dividirá en cuatro partes; la primera consistirá en la preparación de los nodos a nivel *hardware* y *software*, con el fin de replicar el Servidor Web original. Una vez que tenemos ambos nodos configurados, se llevará a cabo su conexión, de tal forma que se establezca una comunicación entre ambos nodos. La segunda parte consistirá en el diseño del *software* de Alta Disponibilidad, que consistirá en la realización de diagramas de flujo que den pauta a la tercera parte: la codificación o desarrollo del *software*, para esta parte se determinarán los lenguajes y las herramientas que optimicen los recursos y faciliten el desarrollo. La cuarta y última parte consistirá en la instalación del *software* de Alta Disponibilidad previamente probado en el ambiente de desarrollo.

[11] Ver Capítulo 4, sección 4.6

5.4.1 PREPARACIÓN DE LOS NODOS DEL CLUSTER

El siguiente procedimiento muestra la forma en que se instala y configura el Nodo 1 del *Cluster* de Alta Disponibilidad, sin embargo de la misma forma se instala y configura el Nodo 2, por tanto evitaré redundar esta información y sólo haré hincapié en las particularidades de cada nodo.

Identificación en la red de cada nodo (Ver Tabla 5.2):

NODO	NOMBRE CANÓNICO	DIRECCIÓN IP	PAPEL EN EL CLUSTER
1	host1.fi-a.unam.mx	192.168.0.1	Web Server(server)
2	host2.fi-a.unam.mx	191.168.0.2	Mirror Server(mirror)

Tabla 5.2 Identificación de los nodos del cluster

El *hardware* de los nodos es el siguiente (Ver Tabla 5.3):

NODO 1	NODO 2
• Procesador *INTEL Pentium* II MMX a 350 MHz • Memoria RAM de 131072 KB • 2 tarjetas de red 10/100 *3COM* Fast Ethernet Link • 2 discos duros: *SAMSUNG* SV1364D (13061 MB) y ST320423A (19532MB)	• Procesador *INTEL Pentium 4* a 2.4 GHz • Memoria RAM de 256 MB • 2 tarjetas de red: *Intel(R)* PRO/1000 Network Connection y *Davicom* DM9102 100 Mbps Full Duplex • 1 disco duro *MAXTOR* (40 GB)

Tabla 5.3 Configuración del *hardware* de los nodos del cluster

5.4.1.1 INSTALACIÓN DEL SISTEMA OPERATIVO

El sistema operativo a instalar es *Linux Fedora Core 2*. Su instalación es relativamente sencilla dado que incorpora una interfase gráfica. Durante este proceso configuramos lo siguiente:

Configuración del lenguaje de instalación: Spanish
Configuración del teclado: Español

Hasta este punto configuramos el idioma de instalación y la distribución del teclado.

Tipo de Instalación: Nueva Personalizada
Particionamiento manual: *Disk Druid*

Para instalar el sistema operativo se debe de preparar los discos duros de instalación, creando las particiones necesarias.

Contamos con dos discos duros:

- */dev/hda* 13061 MB, *SAMSUNG* SV1364D
- */dev/hdc* 19532 MB, ST320423A

Las particiones y los tamaños que se requieren son las siguientes (Ver Tabla 5.4):

PARTICIÓN	TAMAÑO (MB)
swap	260
/casa	2000
/usuarios	2000
/users	200
/home	500
/	resto, 6000

Tabla 5.4 Particiones requeridas

Estas particiones y tamaños son análogos a las del Servidor Web original.

Una vez que se han determinado las particiones y los tamaños, implantaremos la técnica de *Disk Mirroring* para eliminar al disco duro como Punto Simple de Fallo, para ello configuraremos RAID Nivel 1.

La configuración se realiza con *Disk Druid* de una manera muy sencilla, y básicamente consta de los siguientes pasos:

1. Crear una partición de software RAID: En cada disco duro se crea una partición de software RAID de las mismas dimensiones. Por ejemplo en el disco */dev/hda* creamos la partición de software RAID */dev/hda1*, mientras que en el disco duro */dev/hdc* creamos la partición de software RAID */dev/hdc1* (Ver Tablas 5.5 y 5.6, respectivamente)

2. Crear un dispositivo RAID: Una vez que hemos creado las particiones de software RAID, a partir de estas dos creamos el dispositivo RAID correspondiente, asociándolo con un punto de montaje. Continuando con el ejemplo anterior, a partir de las particiones de software RAID */dev/hda1 y /dev/hdc1* creamos el dispositivo */dev/md5* que tiene asociado el punto de montaje / (Raíz). (Ver Tabla 5.7)

/dev/hda1	/dev/md5	Soft RAID	5993 MB
/dev/hda2	/dev/md4	Soft RAID	2102 MB
/dev/hda3	/dev/md1	Soft RAID	2000 MB
/dev/hda4		Extendida	2965 MB
/dev/hda5	/dev/md0	Soft RAID	2000 MB
/dev/hda6	/dev/md2	Soft RAID	502 MB
/dev/hda7		Swap	259 MB
/dev/hda8	/dev/md3	Soft RAID	196.075 MB

Tabla 5.5 Particiones de Software RAID en el disco duro /dev/hda

/dev/hdc1	/dev/md5	Soft RAID	12464 MB
/dev/hdc2	/dev/md4	Soft RAID	2102 MB
/dev/hdc3	/dev/md1	Soft RAID	2000 MB
/dev/hdc4		Extendida	2965 MB
/dev/hdc5	/dev/md0	Soft RAID	2000 MB
/dev/hdc6	/dev/md2	Soft RAID	502 MB
/dev/hdc7		Swap	259 MB
/dev/hdc8	/dev/md3	Soft RAID	196.075 MB

Tabla 5.6 Particiones de Software RAID en el disco duro /dev/hdc

/dev/md0	/casa	ext3	2000.25 MB
/dev/md1	/usuarios	ext3	2000.28 MB
/dev/md2	/home	ext3	502 MB
/dev/md3	/users	ext3	196.075 MB
/dev/md4	/usr	ext3	2102.26 MB
/dev/md5	/	ext3	5992.97 MB

Tabla 5.7 Dispositivos RAID Nivel 1

El siguiente paso es la *instalación del gestor de arranque*, que por defecto es GRUB, al cual se le colocó una contraseña segura para controlar el acceso a la consola si un atacante tuviera la oportunidad de estar físicamente frente al equipo.

Configuración de los dispositivos de red, en Linux se identifican como eth0 y eth1 a la primera y segunda tarjeta de red. (Ver Tabla 5.8):

TARJETA DE RED	CONFIGURACIÓN
eth0	IP = 192.168.0.1 NETMASK = 255.255.255.0 GATEWAY = 192.168.0.254 DNS1 = 132.248.204.1 DNS2 = 132.248.10.2
eth1	Los valores para IP, NETMASK, GATEWAY, DNS1 Y DNS2 son los mismos que eth0

Tabla 5.8 Configuración de las tarjetas de red

Configuración del Firewall: Habilitar el *firewall* permitiendo el tráfico por eth0 y eth1 a través de los puertos 80 (WWW), 22 (SSH) y 5432 (POSGRESQL).

Configuración del idioma del sistema operativo: Spanish (México) y Spanish (Spain), por defecto se selecciona a Spanish (México).

Zona horaria: America/Ciudad_de_México Central Time

Selección de paquetes: En esta sección se selecciona el software a instalar, los paquetes mínimos requeridos son (Ver Tabla 5.9):

Sistema X Windows (Los paquetes que vienen por defecto)
Editores (Sólo Vim_common y Vim_enhanced)
Internet Gráfica (Sólo Mozilla)
Internet basada en texto (Sólo Fechtmail y lynx)
Servidor de correo (Sólo Sendmail-cf)
Herramientas de desarrollo (Los paquetes que vienen por defecto incluyendo gcc-objc y pmake)
Herramientas de administración (Los paquetes que vienen por defecto excepto system-config-soundcard)
Herramientas del sistema (Sólo ethereal, mc, nmap)

Tabla 5.9 Software a instalar

El tamaño de la instalación es de aproximadamente 1,493 MB.

Una vez que se ha decidido que paquetes se instalarán, inicia el proceso de instalación de los mismos.

Autenticación: Se emplea el algoritmo MD5 y el sistema de contraseñas ocultas *shadow*.

Con este último paso prácticamente se da fin a la instalación del sistema operativo.

5.4.1.2 ACTUALIZACIÓN DEL SISTEMA

Para lograr que el sistema este al día, con las últimas actualizaciones, *Fedora Core* cuenta con una herramienta de actualización en línea llamada "up2date". Esta herramienta busca en Internet las últimas actualizaciones y las instala en el sistema operativo. Este proceso de actualización es bastante sencillo y llevará un poco de tiempo en función del tráfico en la red y del número de actualizaciones a descargar. Una vez que se termina de actualizar el sistema, se recomienda reiniciarlo.

5.4.1.3 INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE DE SEGURIDAD

5.4.1.3.1 PORTSENTRY

INSTALACIÓN:

```
[root@tikal seguridad]# tar -zxvf portsentry-1.1.tar.gz
[root@tikal seguridad]# cd portsentry-1.1
```

Editar "*portsentry_config.h*":
```
[root@tikal portsentry-1.1]# vi portsentry_config.h
```

Cambiar:
```
#define CONFIG_FILE "/usr/local/psionic/portsentry/portsentry.conf"
```

Por:
```
#define CONFIG_FILE "/usr/local/doc/portsentry/portsentry.conf"
```

Editar "*Makefile*":
```
[root@tikal portsentry-1.1]# vi Makefile
```

Cambiar:
```
INSTALLDIR = /usr/local/psionic
```

Por:
```
INSTALLDIR = /usr/local/doc
```

Editar "*portsentry.conf*":
```
[root@tikal portsentry-1.1]# vi portsentry.conf
```

Cambiar:
```
# Hosts to ignore
IGNORE_FILE="/usr/local/psionic/portsentry/portsentry.ignore"
# Hosts that have been denied (running history)
HISTORY_FILE="/usr/local/psionic/portsentry/portsentry.history"
# Hosts that have been denied this session only (temporary until next restart)
BLOCKED_FILE="/usr/local/psionic/portsentry/portsentry.blocked"
```

Por:

```
# Hosts to ignore
IGNORE_FILE="/usr/local/doc/portsentry/portsentry.ignore"
# Hosts that have been denied (running history)
HISTORY_FILE="/usr/local/doc/portsentry/portsentry.history"
# Hosts that have been denied this session only (temporary until next restart)
BLOCKED_FILE="/usr/local/doc/portsentry/portsentry.blocked"
```

```
[root@tikal portsentry-1.1]# make linux
[root@tikal portsentry-1.1]# make install
```

CONFIGURACIÓN:

```
[root@tikal portsentry-1.1]# cd /usr/local/doc/portsentry/
```

Editar "*portsentry.conf*":

```
[root@tikal portsentry]# vi portsentry.conf
```

Localizamos las siguientes líneas:

```
# Default TCP ident and NetBIOS service
ADVANCED_EXCLUDE_TCP="113,139"
# Default UDP route (RIP), NetBIOS, bootp broadcasts.
ADVANCED_EXCLUDE_UDP="520,138,137,67"
```

Y las editamos de esta forma:

```
# Default TCP ident and NetBIOS service
ADVANCED_EXCLUDE_TCP="113,139,135,443"
# Default UDP route (RIP), NetBIOS, bootp broadcasts.
ADVANCED_EXCLUDE_UDP="520,138,137,67,135,68"
```

Quitamos el comentario y editamos las siguientes líneas:

```
# iptables support for Linux
KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP; /sbin/service iptables save"
```

5.4.1.3.2 SCANLOGD

```
[root@tikal seguridad]# rpm -ivh scanlogd-2.2-1.5.i386.rpm
```

Al instalar el RPM, scanlogd agrega un nuevo usuario llamado "*firewall*".

Editar el archivo *"/etc/syslog.conf"* y agregar estas líneas al final del archivo:

```
#Bitacora de scanlogd
daemon.alert            /var/log/escaneos
```

/var/log/escaneos es la bitácora de actividad de scanlogd.

5.4.1.3.3 LOGCHECK

INSTALACIÓN:

```
[root@tikal seguridad]# tar -zxvf  logcheck-1.1.1.tar.gz
[root@tikal seguridad]# cd logcheck-1.1.1
```

Editar el archivo "*Makefile*":

```
[root@tikal logcheck-1.1.1]# vi Makefile
```

Cambiar:

```
INSTALLDIR = /usr/local/etc
INSTALLDIR_SH = /usr/local/etc
TMPDIR = /usr/local/etc/tmp
```

Por:

```
# This is where keyword files go.
INSTALLDIR = /usr/local/doc/logckeck
INSTALLDIR_SH = /usr/local/doc/logckeck
TMPDIR = /usr/local/doc/logckeck/tmp

[root@tikal logcheck-1.1.1]# mkdir /usr/local/doc/logckeck
[root@tikal logcheck-1.1.1]# make linux
```

CONFIGURACIÓN:

```
[root@tikal portsentry]# cd /usr/local/doc/logckeck/
```

Editar el archivo "*logcheck.sh*":

```
[root@tikal logckeck]# vi logcheck.sh
```

Editar la última parte del script:

```
# If there are results, mail them to sysadmin
if [ "$ATTACK" -eq 1 ]; then
      cat $TMPDIR/checkreport.$$ | $MAIL -s "$HOSTNAME $DATE ACTIVE SYSTEM ATTACK!"
$SYSADMIN
elif [ "$FOUND" -eq 1 ]; then
      cat $TMPDIR/checkreport.$$ | $MAIL -s "$HOSTNAME $DATE system check" $SYSADMIN
fi
# Clean Up
rm -f $TMPDIR/check.$$ $TMPDIR/checkoutput.$$ $TMPDIR/checkreport.$$

# If there are results, mail them to sysadmin
date=`date |awk '{printf("%s_%s_%s",$2,$3,$4)}'`

if [ "$ATTACK" -eq 1 ]; then
#       cat $TMPDIR/checkreport.$$ | $MAIL -s "$HOSTNAME $DATE ACTIVE SYSTEM ATTACK!"
$SYSADMIN
       cat $TMPDIR/checkreport.$$ > /var/log/bits/bits_logcheck/logcheck_$date
elif [ "$FOUND" -eq 1 ]; then
#       cat $TMPDIR/checkreport.$$ | $MAIL -s "$HOSTNAME $DATE system check" $SYSADMIN
       cat $TMPDIR/checkreport.$$ > /var/log/bits/bits_logcheck/logcheck_$date
fi
```

```
# Clean Up
rm -f $TMPDIR/check.$$ $TMPDIR/checkoutput.$$ $TMPDIR/checkreport.$$
```

[root@tikal logckeck]# mkdir -p /var/log/bits/bits_logcheck

5.4.1.3.4 PUESTA A PUNTO DE LAS HERRAMIENTAS

PUESTA EN MARCHA DE PORTSENTRY Y SCANLOGD:

Editar el archivo *"/etc/init.d/network":*
[root@tikal logckeck]# vi /etc/init.d/network

Agregar las siguientes líneas debajo de "start)":

```
case "$1" in
 start)
      cd /usr/local/doc/portsentry
      ./portsentry -atcp
      ./portsentry -aucp
      /usr/sbin/scanlogd
```

Agregar las siguientes líneas debajo de "stop)":

```
 stop)
      killall portsentry
      killall scanlogd
```

Reiniciar el demonio que controla la tarjeta de red:
[root@tikal logckeck]# /etc/init.d/network restart

PUESTA EN MARCHA DE LOGCKECK:

Agregar al archivo de crones las siguientes líneas:

```
0 * * * 1-5 /bin/sh /usr/local/doc/logcheck/logcheck.sh
0 13,23 * * 6,0 /bin/sh /usr/local/doc/logcheck/logcheck.sh
```

5.4.1.4 CONFIGURACIÓN POSTINSTALACIÓN

5.4.1.4.1 CONFIGURACIÓN DE LOS ARCHIVOS DEL DIRECTORIO /etc

Editar el archivo *motd* y colocar el siguiente banner:
[root@tikal root]# vi /etc/motd
BIENVENIDO A NODO1

Editar el archivo *issue* y colocar el siguiente banner:
[root@tikal root]# vi /etc/issue
BIENVENIDO A NODO1

Editar el archivo *issue.net* y colocar el siguiente banner:
[root@tikal root]# vi /etc/issue.net
BIENVENIDO A NODO1

Editar el archivo *hosts* y colocar las siguientes líneas:

```
[root@tikal root]# vi /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          host1.fi-a.unam.mx host1 localhost.localdomain localhost
192.168.0.1        host1.fi-a.unam.mx host1 nodo1
192.168.0.2        host2.fi-a.unam.mx host2 nodo2
```

Editar el archivo *hosts.allow* y colocar la siguiente línea:

```
hosts.allow
[root@tikal etc]# vi /etc/hosts.allow
#
# hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
ALL: 192.168.0.1
```

Editar el archivo *fstab* y editar las líneas que hacen referencia a */casa, /home, /users y /usuarios* agregando las opciones nodev,noexec,nosuid.

```
[root@tikal root]# vi /etc/fstab
/dev/md0          /casa          ext3   defaults,nodev,noexec,nosuid     1 2
/dev/md2          /home          ext3   defaults,nodev,noexec,nosuid     1 2
/dev/md3          /users         ext3   defaults,nodev,noexec,nosuid     1 2
/dev/md1          /usuarios      ext3   defaults,nodev,noexec,nosuid     1 2
```

Crear el archivo *cron.allow* colocando a root como único usuario válido para ejecutar el demonio cron:

```
[root@tikal root]# vi /etc/cron.allow
root
```

Editar el archive *grub.conf* y eliminar la version del *kernel* del sistema operativo en la línea title y establecer el valor de timeout en 2:

```
[root@tikal etc]# vi grub.conf
timeout=2
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
password --md5 $1$siXM43DR$x0lHO60KQU5EVuydRuER41
title NODO1
      root (hd0,0)
      kernel /boot/vmlinuz-2.6.5-1.358 ro root=/dev/md5 rhgb quiet
      initrd /boot/initrd-2.6.5-1.358.img
```

Ejecutar */sbin/grub* para que los cambios tomen forma:

```
[root@tikal etc]# /sbin/grub
```

Una vez que entremos en el prompt de grub ejecutar:

```
grub> quit
```

5.4.1.4.2 CONFIGURACIÓN DE SSH

Editar el archivo */etc/ssh/sshd_config* y establecer los siguientes valores:

```
Port 22
Protocol 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
LogLevel INFO
LoginGraceTime 20
PermitRootLogin yes
StrictModes yes
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys
RhostsAuthentication no
IgnoreRhosts yes
HostbasedAuthentication no
IgnoreUserKnownHosts no
PasswordAuthentication yes
PermitEmptyPasswords no
X11Forwarding no
PrintMotd yes
PrintLastLog yes
KeepAlive yes
UseLogin no
UsePrivilegeSeparation yes
PermitUserEnvironment no
Compression yes
Banner /etc/issue
```

5.4.1.4.3 CAMBIANDO PERMISOS A COMANDOS

Existen comandos que habrá que cambiarle los permisos que por defecto tienen establecidos, de tal forma que no puedan ser ejecutamos por otro usuario que no sea el administrador. Los permisos que estableceremos para dichos comandos son 0700. Los comandos son los siguientes (Ver Tabla 5.10):

nmap
rpcinfo
vmstat
mount
netreport
pub
traceroute
gpasswd
rpc*
ifconfig
tcpdump
wall
lsattr
arch
chgrp
chvt
gcc
gcc++
make
flex
lex
bison
yacc
perl

Tabla 5.10 Lista de comandos

5.4.1.4.4 CONFIGURACIÓN DE LOS SERVICIOS

Mediante la herramienta setup, configurar cada runlevel de acuerdo a la Tabla 5.11:

RUNLEVEL	SERVICIOS
Init 3	• crond • gpm • iptables • mdmonitor • mdmpd • network • random • sshd • syslog
Init 2	• crond • gpm • iptables • mdmonitor • mdmpd • network • random • sshd • syslog
Init 1	Ningún servicio

Tabla 5.11 Servicios configurados en cada runlevel

5.4.1.4.5 USUARIOS E INFORMACIÓN

En éste rubro, para tratar de replicar al Servidor Web original, se ha realizado una copia de la información de los usuarios así como del sitio Web principal. Para ello copiamos el contenido de los directorios */casa, /usuarios, /home, /users y /usr/local/pgsql/data*, así como el contenido de los archivos */etc/group, /etc/passwd y /etc/shadow*.

Para realizar las pruebas en un entorno seguro, se ejecutó el siguiente script que bloquea las cuentas:

```
[root@tikal root]# vi block.bash
for i in `cat usuarios`
do
      passwd -l $i
      echo "Bloqueando la cuenta $i..."
done
```

En el archivo usuarios se cuenta con los nombres de usuarios obtenidos del archivo */etc/passwd*.

5.4.1.4.6 ESTABLECIENDO LAS CUOTAS A LOS USUARIOS

Editar el archivo */etc/fstab* y en la columna de opciones agregar la opción usrquota en las líneas relativas a */home, /users y /usuarios*:

```
[root@tikal root]# vi /etc/fstab
/dev/md2        /home       ext3  defaults,nodev,noexec,nosuid,usrquota    1 2
/dev/md3        /users      ext3  defaults,nodev,noexec,nosuid,usrquota    1 2
/dev/md1        /usuarios   ext3  defaults,nodev,noexec,nosuid,usrquota    1 2
```

Crear el archivo *aquota.user* dentro de los directorios */home, /users y /usuarios*.

Desmontar dichas particiones:

```
[root@tikal root]# umount /usuarios
[root@tikal root]# umount /home
[root@tikal root]# umount /users
```

Montar las unidades:

```
[root@tikal root]# mount -a
```

Levantar el demonio de las cuotas:

```
[root@tikal root]# quotacheck -mf /home/
[root@tikal root]# quotacheck -mf /home/
[root@tikal root]# quotacheck -mf /usuarios/
[root@tikal root]# quotacheck -mf /usuarios/
[root@tikal root]# quotacheck -mf /users/
[root@tikal root]# quotacheck -mf /users/
[root@tikal root]# quotaon /home/
[root@tikal root]# quotaon /usuarios/
[root@tikal root]# quotaon /users/
```

Editar la cuota de un usuario prototipo:

```
[root@tikal root]# edquota protousuario
Disk quotas for user protousuario (uid 20037):
  Filesystem                blocks     soft     hard   inodes     soft     hard
  /dev/md2                       0    10000    10000        0     2000     2000
  /dev/md3                       0    10000    10000        0     2000     2000
  /dev/md1                       0    10000    10000        0     2000     2000
```

Ejecutar el siguiente script para establecer la cuota a todos los usuarios:

```
[root@tikal root]# vi quotas.bash
for i in `cat usuarios`
do
      edquota -p protousuario $i
      echo "Poniendo quota a $i..."
done
[root@tikal root]# bash quotas.bash
```

5.4.1.5 INSTALACIÓN Y CONFIGURACIÓN DE LOS SERVICIOS

5.4.1.5.1 POSTGRESQL

INSTALACIÓN:

En este paso se desempaqueta y se descomprime el código fuente:

```
[root@tikal Webserver]# tar -zxvf postgresql-7.4.3.tar.gz
[root@tikal Webserver]# cd postgresql-7.4.3
```

Configuramos el código fuente de acuerdo a nuestro sistema:

```
[root@tikal postgresql-7.4.3]# ./configure
```

Y por último se compila y se instala:

```
[root@tikal postgresql-7.4.3]# make
[root@tikal postgresql-7.4.3]# make install
```

CONFIGURACIÓN:

Se crea al usuario *postgres*, propietario de los procesos del manejador de la base de datos:

```
[root@tikal root]# useradd -g daemon -d /usr/local/pgsql/data postgres
```

Creamos los directorios y archivos adicionales:

```
[root@tikal root]# mkdir /usr/local/pgsql/logs
[root@tikal root]# touch /usr/local/pgsql/logs/server.log
[root@tikal root]# chown -R postgres:daemon /usr/local/pgsql
```

Cambiamos de usuario, puesto que solo postgres puede administrar el manejador de base de datos:

```
[root@tikal root]# su - postgres
```

Iniciamos la base de datos:

```
[postgres@tikal data]$ /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

Levantamos el demonio de pgsql:
[postgres@tikal data]$ /usr/local/pgsql/bin/postmaster -i -S -D /usr/local/pgsql/data >> /usr/local/pgsql/logs/server.log

Editamos el archivo */etc/profile* y agregamos la variables de entorno para PostgreSQL :
[root@tikal root]# vi /etc/profile

```
PGDATA="/usr/local/pgsql/date"
PGLIB="/usr/local/pgsql/lib"
LD_LIBRARY_PATH="/usr/local/pgsql/bin"
PATH="$PATH:/usr/local/pgsql/bin"

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC PGDATA PGLIB
LD_LIBRARY_PATH
```

Agregamos las líneas correspondientes a "*local*" y a "*host*" al archivo */usr/local/pgsql/data/pg_hba.conf*, estas líneas llevan el control de usuarios del manejador de base de datos.

Establecemos una contraseña de acceso a la metabase de datos de administración *template1* con acceso sólo para *postgres*:

```
-bash-2.05b$ psql template1
Welcome to psql 7.4.3, the PostgreSQL interactive terminal.
Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit
template1=# alter user postgres with password '&L,5jP';
ALTER USER
```

Una vez hecho esto cometamos las siguientes líneas en el archivo *pg_hba.conf*:

```
-bash-2.05b$ vi pg_hba.conf
#local  all        all                                      trust
#host   all        all        127.0.0.1       255.255.255.255   trust
```

Por ultimo, configuramos la generación de una bitácora de actividad de PostgreSQL:

Editar el archivo */usr/local/pgsql/data/postgresql.conf*:
[root@tikal root]# vi /usr/local/pgsql/data/postgresql.conf

Colocar los siguientes valores en las líneas:

```
syslog = 2                  # range 0-2; 0=stdout; 1=both; 2=syslog
syslog_facility = 'LOCAL0'
syslog_ident = 'postgres'
log_connections = on
```

Creamos el archivo */etc/logrotate/postgres* con el siguiente contenido:

```
[root@tikal root]# vi /etc/logrotate.d/postgres
/var/log/postgresql.log {
        compress
        rotate 2
        size=10000k
        create 0664 postgres postgres
        daily
        postrotate
                /usr/bin/killall -HUP syslogd
        endscript
}
```

Agregar las siguientes líneas al final del archivo */etc/syslog.conf* :

```
[root@tikal root]# vi /etc/syslog.conf
#Bitacora de PostgreSQL
local0.*                                /var/log/postgresql.log
```

Crear la bitácora y reiniciar los demonios de *syslogd y pgsql*:

```
[root@tikal root]# touch /var/log/postgresql.log
[root@tikal root]# chown postgres:daemon /var/log/postgresql.log
[root@tikal root]# /etc/init.d/syslog restart
[root@tikal root]# /bin/su - postgres
-bash-2.05b$ /usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data reload
```

Agregamos las líneas necesarias para levantar automáticamente el servicio de base de datos en el archivo */etc/rc.local*:

```
[root@tikal root]# vi /etc/rc.local
su -c '/usr/local/pgsql/bin/postmaster -i -S -D /usr/local/pgsql/data 2>&1 > /va
r/log/postgresql.log' postgres &
```

Como parte de una configuración segura, borramos el directorio *doc* de la instalación de PosgreSQL:

```
[root@tikal root]# cd /usr/local/pgsql/
[root@tikal pgsql]# ls
bin  data  doc  include  lib  logs  man  share
[root@tikal pgsql]# rm -rf doc/
```

5.4.1.5.2 APACHE HTTP SERVER

INSTALACIÓN:

Desempaquetamos y descomprimimos el código fuente:

```
[root@tikal Webserver]# tar -zxvf httpd-2.0.50.tar.gz
[root@tikal Webserver]# cd httpd-2.0.50
```

Configuramos el código fuente de acuerdo a nuestro sistema:

```
[root@tikal httpd-2.0.50]# ./configure --prefix=/usr/local/apache --enable-so --enable-rewrite
```

Compilamos e instalamos:

```
[root@tikal httpd-2.0.50]# make
[root@tikal httpd-2.0.50]# make install
```

CONFIGURACIÓN:

Creamos al usuario *webadmin*, propietario de los procesos del demonio *httpd*:
[root@tikal httpd-2.0.50]# groupadd webadmin
[root@tikal httpd-2.0.50]# useradd -g webadmin webadmin

Editamos el archivo configuración de Apache:
[root@tikal root]# vi /usr/local/apache/conf/httpd.conf

Editamos las siguientes directivas:
User webadmin
Group webadmin
ServerAdmin webadmin@host1.fi-a.unam.mx
UseCanonicalName On

Establecemos el directorio donde se alojará el sitio principal:
DocumentRoot "/casa/httpd/html"

Caracterizamos al directorio */*, denegando todo acceso desde el Web:
```
<Directory />
  Options None
  AllowOverride None
  Order Deny,Allow
   Deny from all
</Directory>
```

Caracterizamos *DocumentRoot*:
```
<Directory "/casa/httpd/html">
  Options SymLinksIfOwnerMatch IncludesNoExec
  <LimitExcept GET POST>
    Order Deny,Allow
     Deny from all
  </LimitExcept>
</Directory>
```

Indicamos que grupos de usuarios pueden tener su página Web:
UserDir /home/unicafi/*/WWW /usuarios/administrativos/*/WWW /usuarios/ayudprof/*/WWW /usuarios/externos/*/WWW /usuarios/facultad/*/WWW /usuarios/funcionarios/*/WWW /usuarios/investigadores/*/WWW /usuarios/profesores/*/WWW /usuarios/proyecto/*/WWW /usuarios/tecacad/*/WWW

Deshabilitamos esta característica al *home directory* de *root*:
UserDir disabled root

Caracterizamos cada *UserDir*:
```
<Directory /home/unicafi/*/WWW>
      Options SymLinksIfOwnerMatch IncludesNoExec
      AllowOverride None
      <LimitExcept GET POST>
            Order Deny,Allow
             Deny from all
      </LimitExcept>
</Directory>
```

```
<Directory /usuarios/administrativos/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/ayudprof/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/externos/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/facultad/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/funcionarios/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/investigadores/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>
```

```
<Directory /usuarios/profesores/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/proyecto/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>

<Directory /usuarios/tecacad/*/WWW>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    <LimitExcept GET POST>
        Order Deny,Allow
        Deny from all
    </LimitExcept>
</Directory>
```

Caraterizamos los directorios que tienen un fin en específico:

```
Alias /intranet "/casa/intranet"
<Directory /casa/intranet>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride AuthConfig
    Order Deny,Allow
    Deny from all
    Allow from 192.168.0.
    Allow from 192.168.1.
    Allow from 192.168.2.
</Directory>

Alias /inscripciones "/users/inscripciones/altasystem"
<Directory /users/inscripciones>
    Options SymLinksIfOwnerMatch IncludesNoExec
    AllowOverride None
    Order Deny,Allow
    Deny from All
    Allow from 192.168.0.34
    Allow from 192.168.0.65
    Allow from 192.168.0.26
    Allow from 192.168.1.77
    Allow from 192.168.1.68
</Directory>

Alias /estadisticas "/users/estadisticas"
<Directory "/users/estadisticas">
    AllowOverride None
    Options None
```

```
    Order deny,allow
    Deny from all
    Allow from 192.168.0.1
</directory>
```

Editamos las siguientes directives colocando los valores correspondientes:

```
DirectoryIndex index.html index.htm index.php index.php3 index.php4
HostnameLookups On
ServerTokens Prod
```

Comentamos las siguientes líneas:

```
#Alias /icons/ "/usr/local/apache/icons/"
#<Directory "/usr/local/apache/icons">
#   Options Indexes MultiViews
#   AllowOverride None
#   Order allow,deny
#   Allow from all
#</Directory>
#AliasMatch ^/manual(?:/(?:de|en|es|fr|ja|ko|ru))?(/.*)?$ "/usr/local/apache/manual$1"
#<Directory "/usr/local/apache/manual">
#   Options Indexes
#   AllowOverride None
#   Order allow,deny
#   Allow from all

#   <Files *.html>
#      SetHandler type-map
#   </Files>
#   SetEnvIf Request_URI ^/manual/de/ prefer-language=de
#   SetEnvIf Request_URI ^/manual/en/ prefer-language=en
#   SetEnvIf Request_URI ^/manual/es/ prefer-language=es
#   SetEnvIf Request_URI ^/manual/fr/ prefer-language=fr
#   SetEnvIf Request_URI ^/manual/ja/ prefer-language=ja
#   SetEnvIf Request_URI ^/manual/ko/ prefer-language=ko
#   SetEnvIf Request_URI ^/manual/ru/ prefer-language=ru
#   RedirectMatch 301 ^/manual(?:/(de|en|es|fr|ja|ko|ru)){2,}(/.*)?$ /manual/$1$2
#</Directory>
#ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"

#<IfModule mod_cgid.c>
#
# Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
# for setting UNIX socket for communicating with cgid.
#
#Scriptsock        logs/cgisock
#</IfModule>
#<Directory "/usr/local/apache/cgi-bin">
#   AllowOverride None
#   Options None
#   Order allow,deny
#   Allow from all
#</Directory>
```

```
#IndexOptions FancyIndexing VersionSort

#ddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

#AddIconByType (TXT,/icons/text.gif) text/*
#AddIconByType (IMG,/icons/image2.gif) image/*
#AddIconByType (SND,/icons/sound2.gif) audio/*
#AddIconByType (VID,/icons/movie.gif) video/*

#AddIcon /icons/binary.gif .bin .exe
#AddIcon /icons/binhex.gif .hqx
#AddIcon /icons/tar.gif .tar
#AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
#AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
#AddIcon /icons/a.gif .ps .ai .eps
#AddIcon /icons/layout.gif .html .shtml .htm .pdf
#AddIcon /icons/text.gif .txt
#AddIcon /icons/c.gif .c
#AddIcon /icons/p.gif .pl .py
#AddIcon /icons/f.gif .for
#AddIcon /icons/dvi.gif .dvi
#AddIcon /icons/uuencoded.gif .uu
#AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
#AddIcon /icons/tex.gif .tex
#AddIcon /icons/bomb.gif core

#AddIcon /icons/back.gif ..
#AddIcon /icons/hand.right.gif README
#AddIcon /icons/folder.gif ^^DIRECTORY^^
#AddIcon /icons/blank.gif ^^BLANKICON^^
#DefaultIcon /icons/unknown.gif
#ReadmeName README.html
#HeaderName HEADER.html
#IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
AddType application/x-httpd-php .php .phtml .php3 .php4
#<IfModule mod_ssl.c>
#    Include conf/ssl.conf
#</IfModule>
```

Por ultimo creamos *DocumentRoot* e iniciamos el demonio de *httpd*:

```
[root@tikal root]# mkdir /casa/httpd/html
[root@tikal root]# chmod -R 755 /casa/
[root@tikal root]# /usr/local/apache/bin/apachectl start
```

Agregamos las líneas necesarias para levantar automáticamente el servicio Web en el archivo */etc/rc.local*:

```
[root@tikal root]# vi /etc/rc.local
/usr/local/apache/bin/apachectl start
```

5.4.1.5.3 PHP (HYPERTEXT PREPROCESSOR)

INSTALACIÓN:

Desempaquetamos y descomprimimos el código fuente:
[root@tikal Webserver]# tar -zxvf php-5.0.0.tar.gz

Configuramos el código fuente acorde a nuestro sistema:
[root@tikal Webserver]# cd php-5.0.0
[root@tikal php-5.0.0]# ./configure --prefix=/usr/local/php --with-config-file-path=/usr/local/php --with-apxs2=/usr/local/apache/bin/apxs --with-pgsql=/usr/local/pgsql

Compilamos e instalamos:
[root@tikal php-5.0.0]# make
[root@tikal php-5.0.0]# make install

CONFIGURACIÓN:

Hacer una copia del archivo *php.ini-recommend* al directorio de instalación:
[root@tikal php-5.0.0]# cp php.ini-recommended /usr/local/php/php.ini

Editamos el archivo de configuración de PHP:
[root@tikal php-5.0.0]# vi /usr/local/php/php.ini

Cambiamos el valor de la variable *register_globals* por el de *On*:
; You should do your best to write your scripts so that they do not require
; register_globals to be on; Using form variables as globals can easily lead
; to possible security problems, if the code is not very well thought of.
register_globals = On

Editamos el archivo */usr/local/apache/conf/httpd.conf*:
[root@tikal php-5.0.0]# vi /usr/local/apache/conf/httpd.conf

En la sección de *Dynamic Shared Object (DSO) Support* colocamos la siguiente línea:
LoadModule php5_module modules/libphp5.so

Y en Section 2: 'Main' server configuration, en la parte relacionada con las aplicaciones agregamos la siguiente línea:
#
The
If the AddEncoding directives above are commented-out, then you
probably should define those extensions to indicate media types:
#
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php .php .phtml .php3 .php

Por último creamos un archivo de prueba para comprobar la instalación:

```
# vi /casa/httpd/html/phpinfo.php
<?
phpinfo();
?>
```

Reiniciamos el demonio de *httpd*:

```
[root@tikal pgsql]# /usr/local/apache/bin/apachectl restart
```

Y en un navegador probamos la siguiente URL: http://192.168.0.1/phpinfo.php

Y el recurso devuelvo por el servidor es una página dinámica con toda la información referente a la instalación de los servicios Web.

5.4.1.6 PARTICULARIDADES DEL NODO 2

Además de las particularidades mencionadas como la identificación en la red y el tipo de *hardware* con el que cuenta el propio Nodo 2, durante la instalación no se configura RAID Nivel 1, puesto que sólo contamos con un disco duro y no considero necesario llegar a tal nivel de redundancia. En vez de dedicar otro disco duro para este efecto, sería recomendable, contar con al menos una partición lo suficientemente grande para almacenar los respaldos del sistema.

5.4.1.7 CONEXIÓN DE LOS NODOS A LA RED

De acuerdo a la Figura 5.2, podemos observar que la conexión de los Nodos en la red puede tener variantes, en función de los recursos y de la ubicación de los mismos Nodos.

Para garantizar un Nivel de Tolerancia a Fallos, lo ideal es que los Nodos mantengan una distancia considerable, si instalamos un *crossover link* para la comunicación exclusiva entre nodos, con cable UTP Categoría 6, la distancia entre Nodos sería menor a 100m. Mientras se reserva una tarjeta de red para este efecto, tendríamos otras dos tarjetas de red que se conectarían a dos *switches*, respectivamente (Ver Figura 5.2). Esta configuración sería la recomendada, sin embargo, sin llegar a tal nivel de redundancia, la comunicación entre Nodos podría prescindir del *crossover link* y emplear la propia red LAN. Por tanto, esta comunicación deberá estar cifrada, para garantizar la confidencialidad en el medio. De esta forma sólo requeriremos dos tarjetas de red, cada una conectada a un *switch* redundante, y la distancia entre Nodos ya no esta limitada a 100m, sino a la misma configuración de la red LAN.

Considerando el actual escenario, para este Sistema de Alta Disponibilidad, a nivel de *cluster*, la conexión de los Nodos a la red será la siguiente (Ver Figura 5.3):

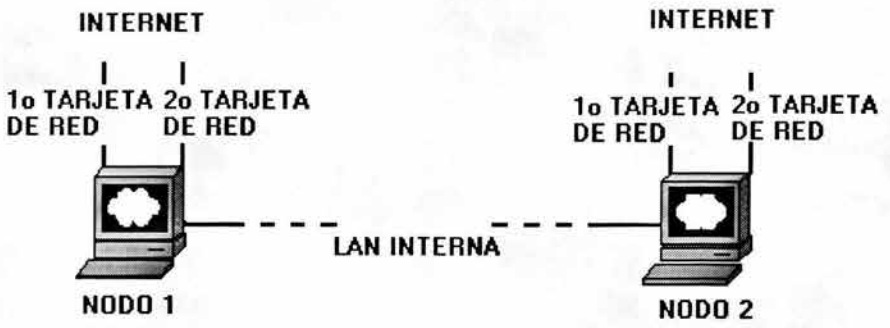

Figura 5.3 Conexión de los Nodos a la red

5.4.2 DISEÑO DEL SOFTWARE DE ALTA DISPONIBILIDAD

5.4.2.1 POSIBLES ESCENARIOS

Los posibles escenarios que pudieran presentarse y las acciones que se tomarán son las siguientes (Ver Tabla 5.12):

ESCENARIO	ACCIONES
Activación del *Cluster* de Alta Disponibilidad	• Nodo 1 inicia el sistema de HA • Nodo 1 registra el hecho en bitácora • Nodo 2 inicia el sistema de HA • Nodo 2 registra el hecho en bitácora
Los 2 Nodos están en correcto funcionamiento	• Nodo 1 tiene el Grupo de Servicios • Nodo 2 respalda la información de manera automática y periódica
El Nodo 1 no presenta conectividad en la red	• Nodo 2 lo detecta y realiza *takeover* apoderándose del Grupo de Servicios • Nodo 2 registra el hecho en bitácora
El Grupo de Servicios en Nodo 1 presenta un fallo	• Nodo 1 lo detecta y notifica a Nodo 2 • Nodo 1 realiza *switchover* • Nodo 1 registra el hecho en bitácora • Nodo 2 se apodera del Grupo de Servicios • Nodo 2 registra el hecho en bitácora
El Nodo 2 no presenta conectividad en la red	• El mismo Nodo 2 lo detecta y lo registra periódicamente en bitácora
Los dos Nodos no presentan conectividad en la red	• Ambos Nodos lo detectan y el que tenga el Grupo de Servicios lo conserva • Ambos Nodos registran el hecho en bitácora
Nodo 2 tiene el Grupo de Servicios y Nodo 1 quiere retomarlos	• Nodo 1 notifica a Nodo 2 • Nodo 1 sincroniza la información • Nodo 2 realiza *switchover* • Nodo 1 se apodera del Grupo de Servicios • Nodo 1 registra el hecho en bitácora • Nodo 2 retoma su estado inicial • Nodo 2 registra el hecho en bitácora
Ambos Nodos están en correcto funcionamiento y se presenta una interrupción planeada que involucra sólo al Nodo 1	• Nodo 1 notifica a Nodo 2 • Nodo 1 realiza *switchover* • Nodo 1 registra el hecho en bitácora • Nodo 2 registra el hecho en bitácora
Ambos Nodos están en correcto funcionamiento y se presenta una interrupción planeada que involucra a ambos Nodos (Detención del *Cluster* de Alta Disponibilidad)	• Se hace la detención del sistema en Nodo 2 (Se desactiva el *Cluster* de Alta Disponibilidad) • Nodo 2 registra el hecho en bitácora • Se hace la detención del sistema en Nodo 1 • Nodo 1 registra el hecho en bitácora

Tabla 5.12 Posibles escenarios y acciones a tomar

5.4.2.2 MODELADO DEL SOFTWARE DE ALTA DISPONIBILIDAD

Una vez que ya hemos determinado los posibles escenarios y las acciones que deberá emprender el *software* de Alta Disponibilidad, el siguiente paso es el modelado. Como parte del modelado, creo conveniente "bautizar" al *software* de Alta Disponibilidad y sus componentes para su fácil identificación. Como hemos visto ya, este software será desarrollado para trabajar con dos nodos en primera instancia, el nodo primario, *Web Server* o Nodo1 recibirá el nombre de "CASTOR", y el nodo secundario, *Mirror Server* o Nodo 2 recibirá el nombre de "POLLUX"; al *software* de Alta Disponibilidad en su conjunto recibirá el nombre de "DIOSCURI".

Estos nombres provienen de la mitología griega, Castor y Pollux, llamados Dioscuros o Dioscuri, héroes griegos hijos de Leda. La historia cuenta como Leda en una misma noche, se unió a Zeus y a su esposo Tindáreo. De estas uniones puso dos huevos. De uno salieron Pollux y Helena, hijos de Zeus. Del otro huevo salieron Castor y Clitemestra, hijos de Tíndáreo.

Castor y Pollux, los gemelos, los hermanos Dioscuros, nunca se separaban el uno del otro en ninguna aventura, y llegaron a ser el orgullo de Esparta. Cástor llegó a ser un famoso soldado y domador de caballos, y Pollux el mejor luchador de su época. Los dos juntos ganaron grandes premios en los Juegos Olímpicos. Los Dioscuros también participaron con Jasón en el viaje de los argonautas a la Cólquida, y lograron aplacar una tempestad que estuvo a punto de hacer naufragar a la nave Argos.

Había dos cazadores llamados Idas y Linceo, que luego fueron argonautas, que un día discutieron con los Dioscuros sobre como unir sus fuerzas para apoderarse de todo el ganado de la Arcadia. Se repartieron éstos echándolo a suertes comiéndose una vaca entre los cuatro. Pollux fue el último en terminar, pues era el más lento comiendo. Pero al final, no conforme con el resultado de la partición, terminaron en pelea, e Idas mató a Castor con una lanza. Pollux se vengó de su hermano, matando a Linceo con su lanza, y Zeus apiadado de su hijo, mató al otro, a Linceo, con un rayo.

Pollux sentía una pena enorme y suplicó a Zeus, no sobrevivir a su querido hermano, ya que el sufrimiento no podía soportarlo. Pero el problema era que Castor era mortal, y Pollux hijo de un dios. Este, fue llevado al cielo, pero rechazó la inmortalidad a no ser que la compartiera con Castor, y Zeus, concedió que ambos pasaran su vida alternativamente en el aire superior y bajo la tierra en Terapne. Y para premiar aún más su amor fraterno, puso sus imágenes entre las estrellas como la constelación de Géminis.

5.4.2.2.1 DIAGRAMAS DE FLUJO

Parte fundamental del modelado es el plasmar los procesos que realizará DIOSCURI ante cada escenario. A continuación muestro los diagramas de flujo para CASTOR, y posteriormente para POLLUX.

5.4.2.2.1.1 CASTOR

La Figura 5.4 muestra el Menú Principal de Operaciones en CASTOR:

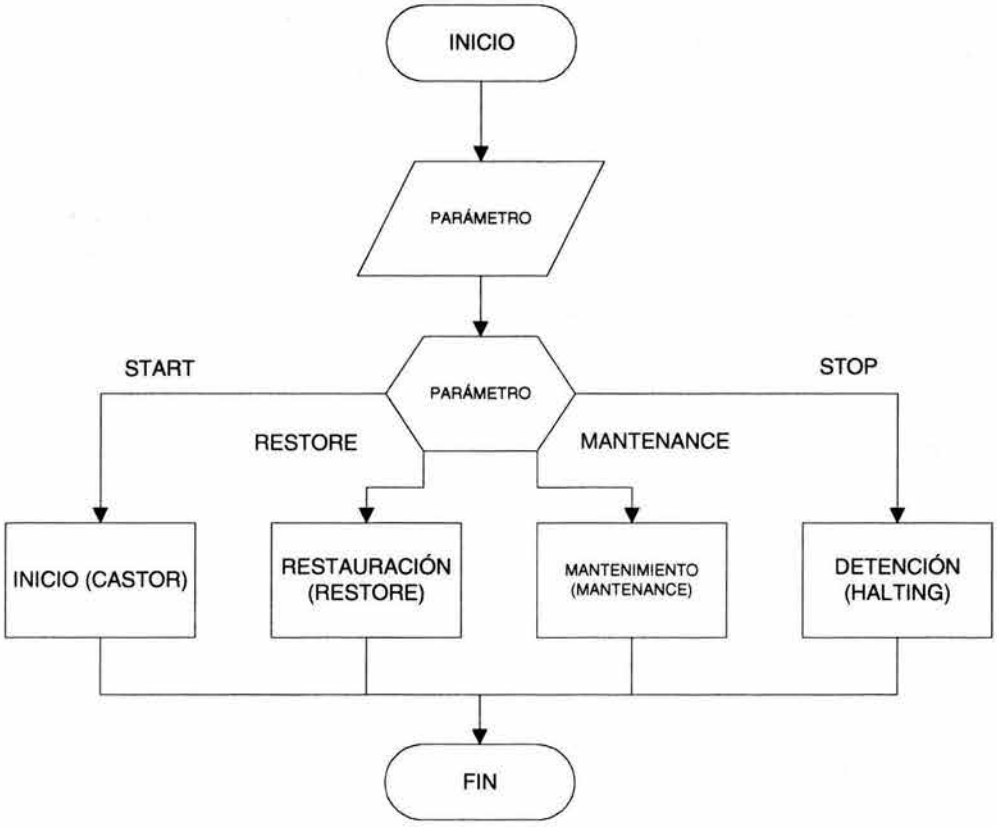

Figura 5.4 Menú Principal de Operaciones en CASTOR

En la figura 5.5 se muestra el diagrama de flujo de CASTOR:

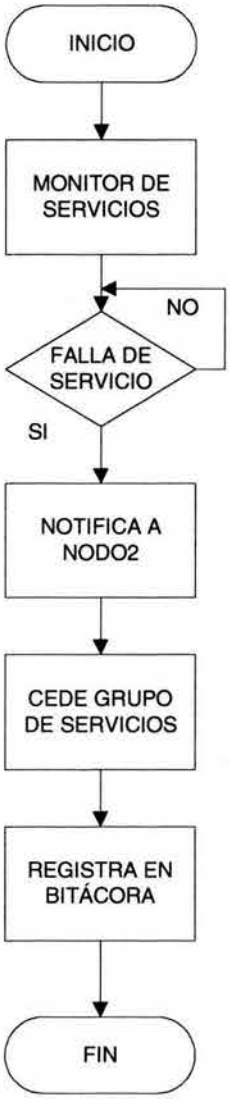

Figura 5.5 Diagrama de Flujo CASTOR

El Diagrama de Flujo para RESTORE es ilustrado en la Figura 5.6:

Figura 5.6 Diagrama de Flujo para RESTORE

Para la función MANTENANCE (Ver Figura 5.7):

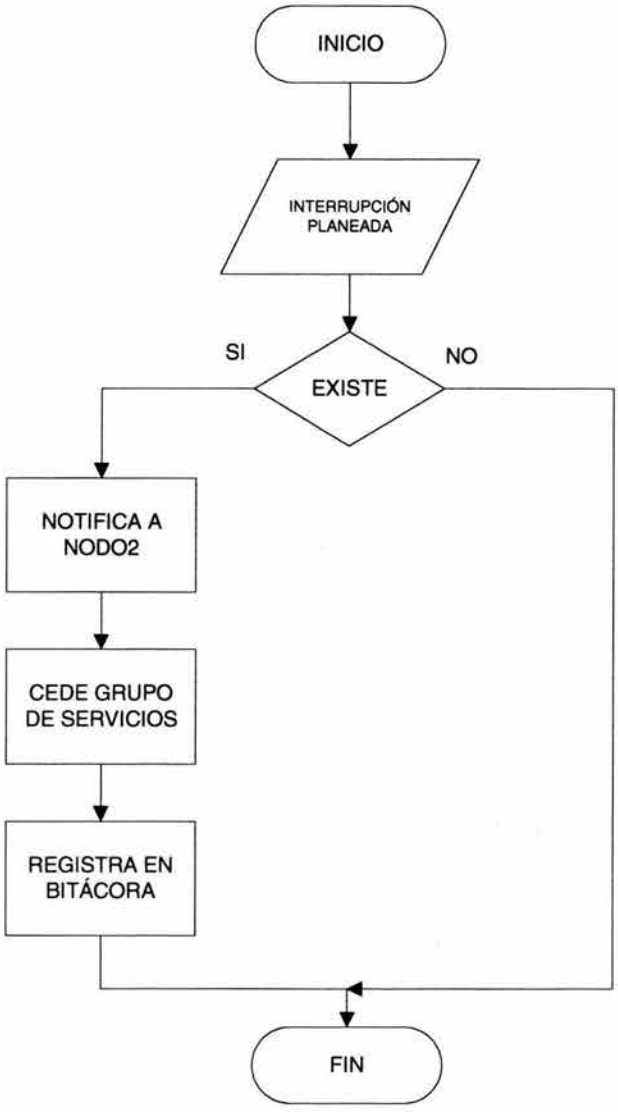

Figura 5.7 Diagrama de Flujo para la función MANTENANCE

Para la función HALTING (Ver Figura 5.8):

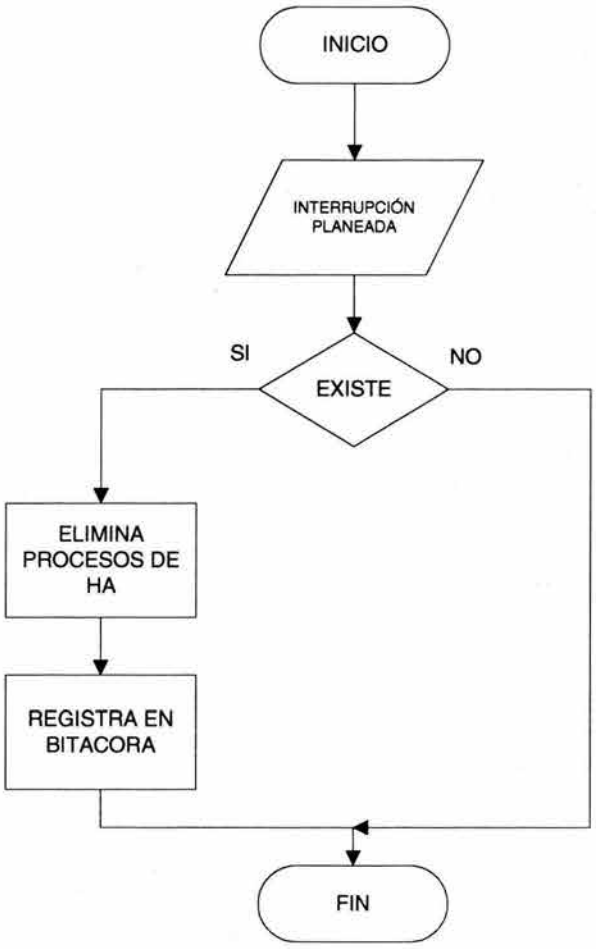

Figura 5.8 Diagrama de Flujo para la función HALTING

5.4.2.2.1.2 POLLUX

En la Figura 5.9 muestro el Menú Principal de Operaciones que controla a POLLUX:

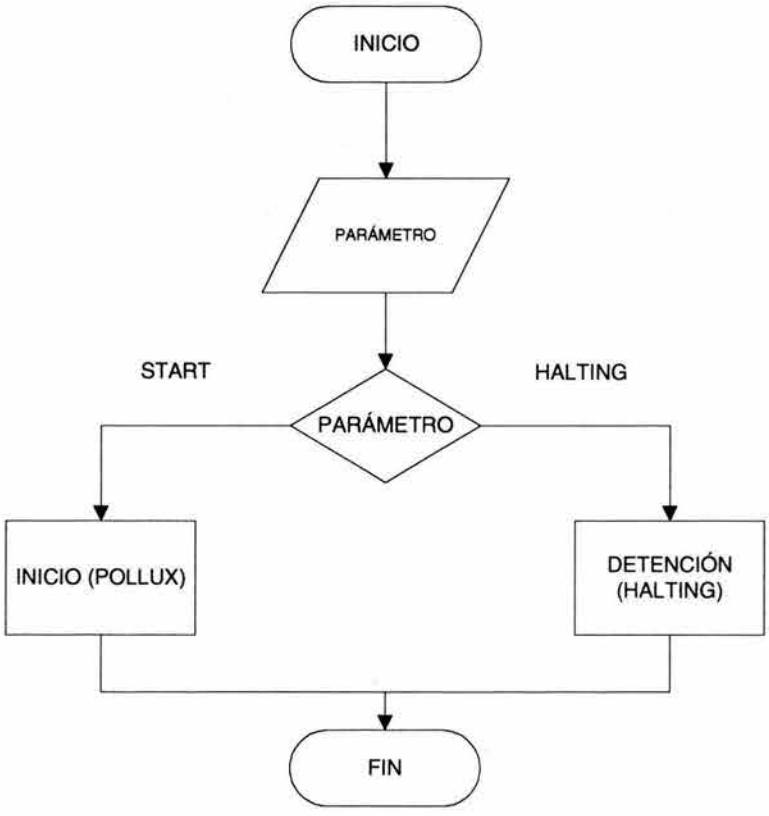

Figura 5.9 Menú Principal de Operaciones para POLLUX

POLLUX y su funcionamiento está ilustrado en la Figura 5.10:

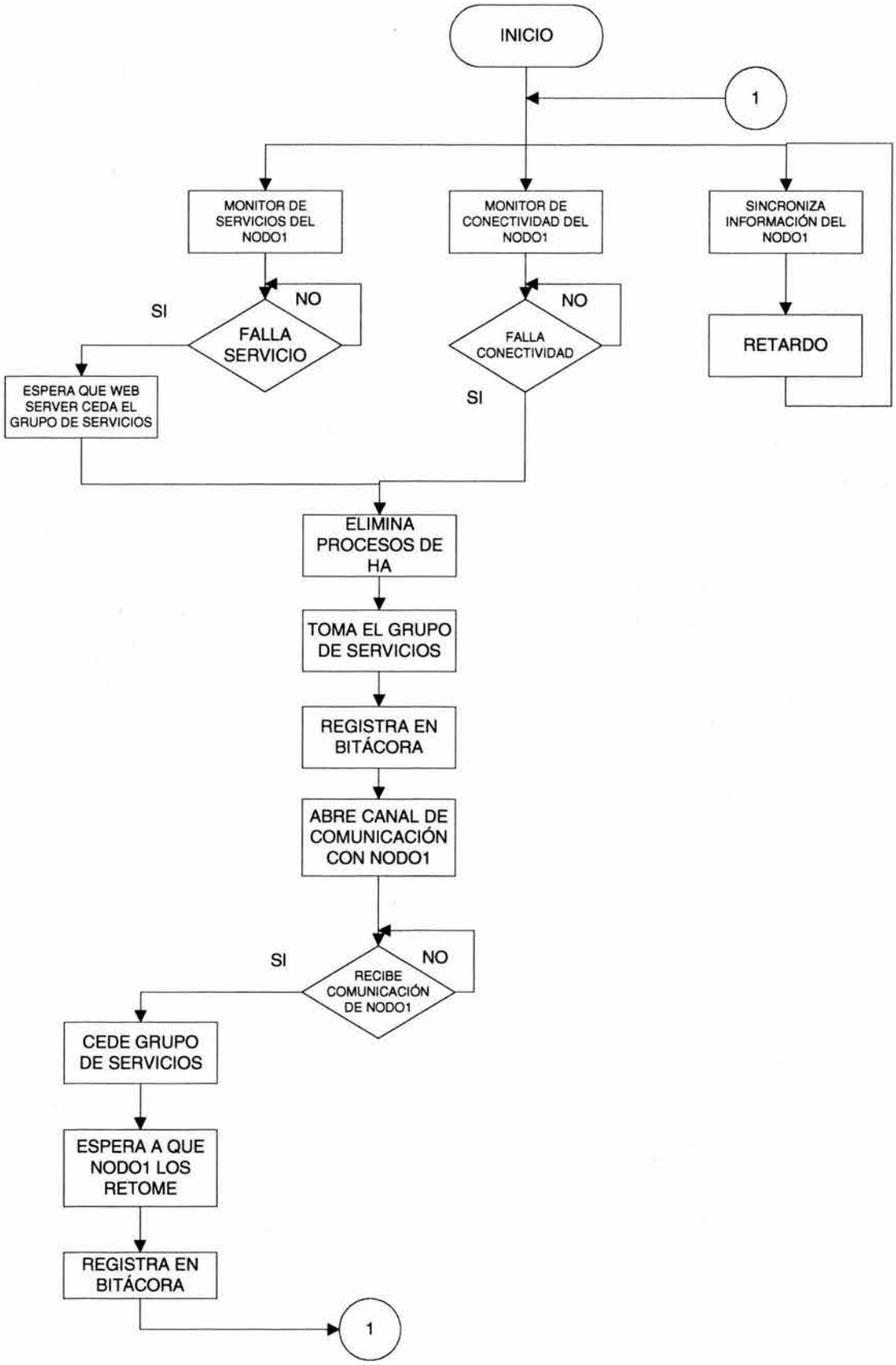

Figura 5.10 Diagrama de Flujo POLLUX

Para el caso del programa HALTING en POLLUX, el diagrama de flujo es el mismo que en el caso de CASTOR, para ello nos remitiremos a la Figura 5.8.

Con esto terminamos el modelado de DIOSCURI, y pasamos a la siguiente parte que es la codificación con base en los diagramas de flujo mostrados.

5.4.3 DESARROLLO DEL SOFTWARE DE ALTA DISPONIBILIDAD

5.4.3.1 HERRAMIENTAS A UTILIZAR EN LA CONSTRUCCIÓN DEL SOFTWARE DE ALTA DISPONIBILIDAD

Para el desarrollo de DIOSCURI, los lenguajes que me permitirán sacarle el máximo provecho a esta nueva herramienta son: Lenguaje C y ShellScript. El primero lo utilizaré para los módulos o las funciones motor del sistema, y ShellScript me servirá para realizar funciones de no tan bajo nivel y como interfase entre el sistema operativo UNIX/Linux y el Sistema de Alta Disponibilidad.

Además de estos lenguajes, haré uso de comandos y utilerías que son instaladas por defecto en toda distribución UNIX/Linux (ping, ifconfig, hostname, entre otros) y de *software* de propósito específico que serán requisito para el correcto funcionamiento del Sistema. El *software* de propósito específico utilizado es el siguiente:

OPENSSL: Es un conjunto de utilerías que básicamente permiten cifrar datos. Su módulo de cifrado Triple DES permitirá que la comunicación entre nodos sea confidencial, puesto que viajará cifrada en el medio.

RSYNC: Básicamente esta herramienta permite no sólo realizar una copia de la información, sino que permite tener una sincronización de ella, de esta forma todos los cambios en la información de un nodo se replican de una forma muy sencilla. Si esta información viajara en formato plano a través del medio, sería inseguro y atentaría contra la confidencialidad de la misma, por ello, RSYNC implementa un mecanismo de cifrado durante su acción.

5.4.3.2 CODIFICACIÓN

Como ya ha sido mencionado, DIOSCURI fue dividido en dos partes: CASTOR y POLLUX; hasta este punto ya conocemos lo que realizará cada parte. Una vez que tenemos ya los algoritmos de funcionamiento, ahora los traduciremos al lenguaje que entiende la computadora. Para el desarrollo he subdividido cada parte de DIOSCURI en pequeños programas o módulos que realizan una función en específico, esto porque hay tareas que como vemos en los diagramas de flujo se repiten tanto en CASTOR como en POLLUX. A continuación muestro como esta constituido CASTOR y como sus módulos están relacionados.

5.4.3.2.1 CÓDIGO FUENTE DE CASTOR

CASTOR está compuesto por los siguientes módulos (Ver Figura 5.11):

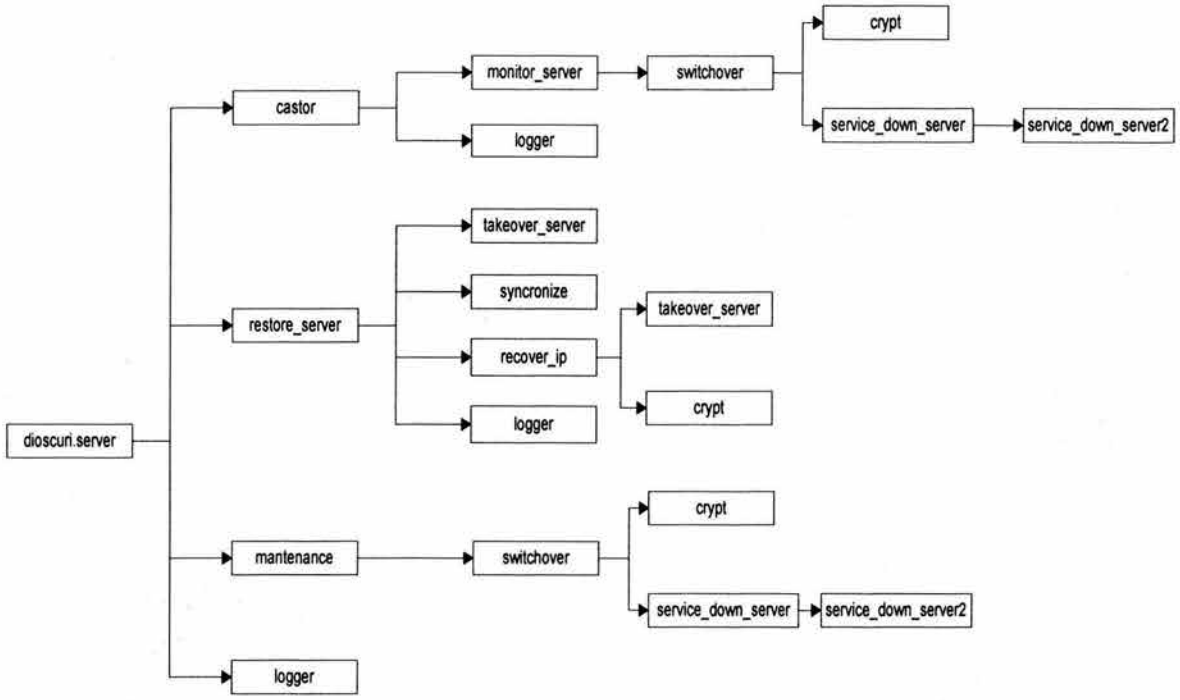

Figura 5.11 Módulos de CASTOR

A continuación haré una descripción de cada módulo e ilustraré el código fuente respectivo. Cabe hacer notar que existen dos archivos de configuración importantes: *dioscuri_config.h* y *castor.conf* sobre los cuales hablaré más adelante.

5.4.3.2.1.1 dioscuri.server

Este módulo tiene la función de ser el menú principal de operaciones, recibe los parámetros *start, restore, mantenance y halting*. La Tabla 5.13 ilustra las acciones que realiza con cada parámetro.

PARÁMETRO	ACCIÓN BÁSICA
start	Ejecuta el módulo *castor*
restore	Ejecuta el módulo *restore_server*
mantenance	Ejecuta el módulo mantenance
halting	Ejecuta una rutina que realiza la detención del sistema, ésta rutina viene incluida en el módulo

Tabla 5.13 Parámetros de funcionamiento

El código fuente de *dioscuri.server* es el siguiente:

```
#!/bin/bash
##################################################
#MENU PRINCIPAL DE OPERACIONES PARA POLLUX
#INICIA O DETIENE EL SISTEMA DE ALTA DISPONIBILIDAD
##################################################

. ../conf/castor.conf
case $1 in

start)          clear
                if [ -f "$INSTALL_PATH_CASTOR""/bin/castor" ]
                then
                echo -n  "Iniciando el Sistema de Alta Disponibilidad...  "
                "$INSTALL_PATH_CASTOR""/bin/castor"&
                "$INSTALL_PATH_CASTOR""/bin/logger" "El Sistema de Alta Disponibilidad ha
sido activado exitosamente."
                echo "HECHO"
                        echo ""
                else
                echo "El Sistema de Alta Disponibilidad no puede ser iniciado."
                echo " El comando \"castor\" no fue encontrado."
                "$INSTALL_PATH_CASTOR""/bin/logger" "El Sistema de Alta Disponibilidad no ha
sido activado debido a que el comando \"castor\" no fue encontrado."
                        echo ""
                fi
                ;;
restore)clear
                echo ""
                if [ -f "$INSTALL_PATH_CASTOR""/bin/restore_server" ]
                then
                echo -n  "Restaurando los servicios en Web Server (nodo 1)..."
                "$INSTALL_PATH_CASTOR""/bin/restore_server"
                echo "HECHO"
                        echo ""
                        sleep 20
                        $0 start
                else
                echo "Los servicios en Web Sever (Nodo 1) no pueden ser restaurados."
                echo " El comando \"restore_server\" no fue encontrado."
                "$INSTALL_PATH_CASTOR""/bin/logger" "Los servicios en Web Server (Nodo 1)
no pueden ser restaurados debido a que el comando \"restore_server\" no fue encontrado."
                        echo ""
                fi
                ;;
mantenance)     clear
                echo "Ha decidido detener el Sistema de Alta Disponibilidad por cuestiones de
mantenimiento."
                echo ""
                echo -n "Proseguir con la detencion? [S/N] "
                read decision
                echo ""
                if [ $decision == S ]
                then
                        echo "Ingrese el motivo por el cual se detendra el Sistema HA: "
```

```
                        echo ""
                        read motivo
                        if [ `echo $motivo|wc -c` -ne 1 ]
                        then
                                "$INSTALL_PATH_CASTOR""/bin/mantenance"
                                cmd=$?
                                clear
                        echo -n "Parando el Sistema HA...  "
                                if [ $cmd -eq 0 ]
                                then
                                        echo "HECHO"
                                        echo ""
                                        "$INSTALL_PATH_CASTOR""/bin/logger"
"INTERRUPCION PLANEADA: $motivo"
                                        "$INSTALL_PATH_CASTOR""/bin/logger" "El Sistema HA
ha sido detenido."
                                else
                                        echo "FALLO"
                                        echo ""
                                        echo "************************************************"
                                        echo "El Sistema HA no esta siendo ejecutado."
                                        echo "************************************************"
                                        echo ""
                                        exit 0
                                fi
                        else
                                echo ""
                                echo
"*****************************************************************************"
                                echo "Debe especificar el motivo por le cual quiere usted detener el
Sistema HA."
                                echo
"*****************************************************************************"
                                echo ""
                        fi
                else
                echo ""
                echo "*********************************************************************"
                echo "El Sistema de Alta Disponibilidad no ha sido modificado."
                echo "*********************************************************************"
                echo ""
        fi
                ;;
halting) clear
                echo "Ha decidido detener el Sistema de Alta Disponibilidad."
                echo ""
                echo "*************************************************************************************"
                echo "ADVERTENCIA: Detenga antes el Sistema de Alta Disponibilidad en el Nodo
2."
                echo "*************************************************************************************"
                echo ""
                echo -n "Proseguir con la detencion? [S/N] "
                read decision
                echo ""
                if [ $decision == S ]
                then
```

```
                        echo "Ingrese el motivo por el cual se detendra el Sistema HA: "
                        echo ""
                        read motivo
                        if [ `echo $motivo|wc -c` -ne 1 ]
                        then
                                killall monitor_server >> /dev/null
                                cmd=$?
                                clear
                        echo -n "Parando el Sistema HA...  "
                                if [ $cmd -eq 0 ]
                                then
                                        echo "HECHO"
                                        echo ""
                                        "$INSTALL_PATH_CASTOR""/bin/logger"
"INTERRUPCION PLANEADA: $motivo"
                                        "$INSTALL_PATH_CASTOR""/bin/logger" "El Sistema HA
ha sido detenido."
                                else
                                        echo "FALLO"
                                        echo ""
                                        echo "************************************************"
                                        echo "El Sistema HA no esta siendo ejecutado."
                                        echo "************************************************"
                                        echo ""
                                        exit 0
                                fi
                        else
                                echo ""
                                echo
"*******************************************************************************"
                                echo "Debe especificar el motivo por le cual quiere usted detener el
Sistema HA."
                                echo
"*******************************************************************************"
                                echo ""
                        fi
                else
                        echo ""
                        echo "*****************************************************************"
                        echo "El Sistema de Alta Disponibilidad no ha sido modificado."
                        echo "*****************************************************************"
                        echo ""
                fi
                ;;
*)              echo "PARAMETROS DE USO: [start|restore|mantenance|halting] "
                exit 0
esac
```

5.4.3.2.1.2 castor

castor básicamente ejecuta los módulos necesarios para monitorear el Grupo de Servicios.

```
#!/bin/bash
#################################################################
#PROGRAMA QUE EJECUTA EL MONITOREO DE SERVICIOS Y VERIFICA
#QUE EL NODO 1 ESTE EN CONECTIVIDAD CON LA RED
#################################################################
. ../conf/castor.conf

if [ -f "$INSTALL_PATH_CASTOR""/bin/monitor_server" ]
then
        "$INSTALL_PATH_CASTOR""/bin/monitor_server"&
else
        echo -n "El Sistema de Alta Disponibilidad no puede ser iniciado."
        echo " El comando \"monitor_server\" no fue encontrado."
        "$INSTALL_PATH_CASTOR""/bin/logger" "El Sistema de Alta Disponibilidad no ha sido
activado debido a que el comando \"monitor_server\" no fue encontrado."
        exit 0
fi
```

5.4.3.2.1.3 restore_server

Ejecuta los módulos necesarios para retomar el Grupo de Servicios después de una interrupción planeada o no planeada.

```
#! /bin/bash
#################################################################################
#PROGRAMA QUE EJECUTA LA TOMA TEMPORAL DE LA IP DEL NODO 2 PARA
COMUNICARSE
#Y OBLIGARLO A UN SWITCHOVER Y ENTREGAR EL GRUPO DE SERVICIOS
#################################################################################
. ../conf/castor.conf

"$INSTALL_PATH_CASTOR""/bin/takeover_server" $IP_NODO2 $HOSTNAME_NODO2
HEARTBEAT_NODO2=`ping -w 1 -c 1 $GATEWAY | grep packets | awk '{print $6}'`
if [ $HEARTBEAT_NODO2 == "100%" ]
then
        "$INSTALL_PATH_CASTOR ""/bin/logger" "Fallo de conectividad despues de tomar la IP
de Nodo 2. No es posible restablecer el servicio."
        echo "No es posible restablecer el servicio. Revise su configuracion de red."
else
        HEARTBEAT_NODO1=`ping -w 1 -c 1 $IP_NODO1 | grep packets | awk '{print $6}'`
        if [ $HEARTBEAT_NODO1 == "100%" ]
        then
                "$INSTALL_PATH_CASTOR ""/bin/logger" "Fallo de conectividad del Mirror Server
(Nodo 2)"
                "$INSTALL_PATH_CASTOR ""/bin/takeover_server" $IP_NODO1
$HOSTNAME_NODO1
                "$INSTALL_PATH_CASTOR ""/bin/logger" "El Web Server (Nodo 1) ha
restablecido el servicio. El Nodo 2 esta fuera de servicio."
```

```
        echo "Se ha restablecido el servicio. Sin embargo el Nodo 2 esta fuera de servicio."
        else
                "$INSTALL_PATH_CASTOR ""/bin/syncronize"
                sleep 10
                "$INSTALL_PATH_CASTOR ""/bin/recover_ip" $IP_NODO1
                "$INSTALL_PATH_CASTOR ""/bin/logger" "El Web Server (Nodo 1) ha
restablecido el servicio."
                echo "Se ha restablecido el servicio."
        fi
        killall monitor_server >> /dev/null
        /etc/init.d/sshd restart >> /dev/null
        /usr/local/apache/bin/apachectl restart >> /dev/null
        su -c '/usr/local/pgsql/bin/postmaster -i -S -D /usr/local/pgsql/data 2>&1 >
/var/log/postgresql.log' postgres& >> /dev/null
fi
```

5.4.3.2.1.4 mantenance

Este módulo es llamado cuando el sistema entrará en un periodo de mantenimiento, cediendo el Grupo de Servicios al Nodo 2:

```
/*#########################################################################*/
/*PROGRAMA QUE EN CASO DE MANTENIMIENTO                                    */
/*REGISTRA EL HECHO EN BITACORA Y EJECUTA SWITCHOVER                       */
/*#########################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <time.h>
#include <syslog.h>
#include "dioscuri_config.h"
#define RETARDO 10
int crea_socket(int);
struct sockaddr_in nodo;
int fd,sin_size,sshd,httpd,pgsql;
int main(int argc, char *argv[])
{
        int demonio;
    demonio=fork();
    if(demonio < 0)
    {
        perror(errno);
        return -1;
    }
    else if (demonio != 0)
    {
                wait();
                execlp(INSTALL_PATH_CASTOR"/bin/switchover","switchover",IP_NODO2,NULL);
        exit(0);
    }
        openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);
        syslog(LOG_EMERG,"El Web Server (Nodo 1) pasa al estado de mantenimiento.");
```

```
        return 0;
}
```

5.4.3.2.1.5 logger

Este módulo es empleado para registrar la actividad del Sistema de Alta Disponibilidad.

```
/*################################################################*/
/*PROGRAMA QUE GENERA UNA ENTRADA EN LA BITACORA DEL SISTEMA      */
/*DE ALTA DISPONIBILIDAD                                          */
/*################################################################*/

#include <stdio.h>
#include <syslog.h>

main(argc,argv)
int argc;
char *argv[];
{
        char mensaje[100];
        strcpy(mensaje,argv[1]);
    openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);
    syslog(LOG_EMERG,mensaje);
}
```

5.4.3.2.1.6 monitor_server

Se encarga propiamente de realizar el monitoreo del Grupo de Servicios en el Nodo, en caso de falla, cede el Grupo al Nodo alterno.

```
/*#############################################################################*/
/*PROGRAMA QUE MONITOREA LOS SERVICIOS DEL WEB SERVER, EN CASO DE FALLA */
/*REGISTRA EL HECHO EN BITACORA Y EJECUTA SWITCHOVER                    */
/*#############################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <time.h>
#include <syslog.h>
#include "dioscuri_config.h"
#define RETARDO 10

int crea_socket(int);
struct sockaddr_in nodo;
int fd,sin_size,sshd,httpd,pgsql;
int main(int argc, char *argv[])
{
        int demonio;
    demonio=fork();
    if(demonio < 0)
```

```
        {
                perror(errno);
                return -1;
        }
        else if (demonio != 0)
        {
                        wait();
                        execlp(INSTALL_PATH_CASTOR"/bin/switchover","switchover",IP_NODO2,NULL);
                exit(0);
        }
        setsid();
        chdir("/");
        umask(0);
           while(1)
           {
                        sshd=crea_socket(PUERTO_SSH);
                        httpd=crea_socket(PUERTO_HTTP);
                        pgsql=crea_socket(PUERTO_PGSQL);
              if((sshd==0)&&(httpd==0)&&(pgsql==0))
              {
                                retardo(RETARDO);
                        }
                        else
                                exit(0);
           }
           return 0;
}

int crea_socket(int PUERTO)
{
           int servicio=0;
           if ((fd=socket(AF_INET, SOCK_STREAM, 0)) == -1 )
           {
                        printf("error en socket()\n");
                        exit(-1);
           }
           nodo.sin_family = AF_INET;
           nodo.sin_port = htons(PUERTO);
           nodo.sin_addr.s_addr = INADDR_ANY;
           bzero(&(nodo.sin_zero),8);
           if(bind(fd,(struct sockaddr*)&nodo,sizeof(struct sockaddr))==-1)
           {
           }
           else
           {
                        openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);
                        syslog(LOG_EMERG,"El puerto %i no tiene un servicio asociado en el Web
Server.",PUERTO);
                        servicio=1;
           }
        close(fd);
           return servicio;
}
```

5.4.3.2.1.7 takeover_server

Este módulo realiza la toma del Grupo de Servicios.

```
/*#####################################################################*/
/*PROGRAMA QUE REALIZA EL TAKEOVER DEL GRUPO DE SERVICIOS DEL NODO 2*/
/*#####################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include "dioscuri_config.h"

main(argc,argv)
int argc;
char *argv[];
{
if(argc==3)
{
        int demonio;
        demonio=fork();
        if(demonio<0)
        {
        perror(errno);
        return -1;
        }
        else if(demonio!=0)
        {
                wait();
                execlp(HOSTNAME,"hostname",argv[2],NULL);
                return 0;
        }

        execlp(IFCONFIG,"ifconfig","eth0",argv[1],"netmask",NETMASK,"broadcast",BROADCAST,
"up",NULL);
        return 0;
}
else
        printf("FALTAN LOS PARAMETROS IP Y HOSTNAME\n");
}
```

5.4.3.2.1.8 syncronize

Realiza la sincronización de la información antes de realizar la toma del Grupo de Servicios propiamente.

```
#!/bin/bash
#########################################################################
#PROGRAMA QUE SINCRONIZA LA INFORMACION DE LOS USUARIOS Y REPORTA EN
#BITACORA DICHO EVENTO
#########################################################################
. ../conf/castor.conf

HEARTBEAT_NODO2=`ping -w 1 -c 1 $GATEWAY 2> /dev/null | grep packets | awk '{print $6}'`

if [ $HEARTBEAT_NODO2 == 0% ]
then
        HEARTBEAT_NODO1=`ping -w 1 -c 1 $IP_NODO1 2> /dev/null | grep packets | awk '{print
$6}'`
        if [ $HEARTBEAT_NODO1 == 0% ]
        then
                for i in $USERS_DIR
                do
                        $RSYNC -ave ssh $IP_NODO1:$i/* $i/  >> /dev/null
                        "$INSTALL_PATH_CASTOR""/bin/logger" "La informacion del directorio $i
esta sincronizada."
                done
          rm -f /usr/local/pgsql/data/postmaster.pid

        else
                "$INSTALL_PATH_CASTOR""/bin/logger" "Fallo de conectividad del Mirror Server
(Nodo 2) al tratar de sincronizar la informacion."
        fi
else
        "$INSTALL_PATH_CASTOR""/bin/logger" "Fallo de conectividad del Web Server (Nodo 1)
al tratar de sincronizar la informacion."
fi
```

5.4.3.2.1.9 recover_ip

Este modulo notifica al Nodo alterno que se tomará el Grupo de Servicios y por último, ejecuta dicha operación.

```
/*#########################################################*/
/*PROGRAMA QUE REALIZA LA RECUPERACION DE LA IP DE NODO 1*/
/*#########################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <errno.h>
#include "dioscuri_config.h"
```

```
int main(int argc, char *argv[])
{
        FILE *mensaje;
        int fd, numbytes,len,bytes_sent,i,demonio;
        char c,buf[MAXDATASIZE], msg[100];
        struct hostent *he;
        struct sockaddr_in server;
        if(argc !=2)
        {
                printf("Uso: %s <Direccion>\n",argv[0]);
                exit(-1);
        }
        demonio=fork();
        if(demonio<0)
        {
                perror(errno);
                return -1;
        }
        else if(demonio!=0)
        {
                wait();
                mensaje=fopen("m_cifrado","r");
                i=0;
                while((c=getc(mensaje))!=EOF)
                {
                        msg[i]=c;
                        i++;
                }
                msg[i+1]='\0';
                fclose(mensaje);
                if((he=gethostbyname(argv[1]))==NULL)
                {
                        printf("gethostbyname() error\n");
                        exit(-1);
                }
                if((fd=socket(AF_INET, SOCK_STREAM, 0))==-1)
                {
                        printf("socket() error\n");
                        exit(-1);
                }
                server.sin_family = AF_INET;
                server.sin_port = htons(PORT);
                server.sin_addr = *((struct in_addr *)he->h_addr);
                bzero(&(server.sin_zero),8);
                if(connect(fd, (struct sockaddr *)&server,sizeof(struct sockaddr))==-1)
                {
                        printf("connect() error\n");
                        exit(-1);
                }
                len=strlen(msg);
                if((bytes_sent=send(fd,msg,len+1,0)) == -1)
                {
                        printf("Error en send() \n");
                        exit(-1);
                }
```

```
                if((numbytes=recv(fd,buf,MAXDATASIZE,0)) == -1)
                {
                        printf("Error en recv() \n");
                        exit(-1);
                }
                buf[numbytes]='\0';
                close(fd);
                retardo(10);

        execlp(INSTALL_PATH_CASTOR"/bin/takeover_server","takeover_server",IP_NODO1,HO
STNAME_NODO1,NULL);
        }
        execlp(INSTALL_PATH_CASTOR"/bin/crypt","crypt","recover",CLAVE,NULL);
        return 0;
}
```

5.4.3.2.1.10 switchover

Este módulo notifica al Nodo alterno que cederá el Grupo de Servicios, posteriormente realiza la detención del servicio de red.

```
/*#################################################################*/
/*PROGRAMA QUE NOTIFICA A NODO 2 QUE CEDERA EL GRUPO DE RECURSOS*/
/*Y REALIZA LA DETENCION DEL SERVICIO DE RED                      */
/*#################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <errno.h>
#include "dioscuri_config.h"

int main(int argc, char *argv[])
{
        FILE *mensaje;
        int fd, numbytes,len,bytes_sent,i,demonio;
        char c,buf[MAXDATASIZE], msg[100];
        struct hostent *he;
        struct sockaddr_in server;
        if(argc !=2)
        {
                printf("Uso: %s <Direccion>\n",argv[0]);
                exit(-1);
        }
        demonio=fork();
        if(demonio<0)
        {
                perror(errno);
                return -1;
        }
        else if(demonio!=0)
        {
                wait();
```

```
                mensaje=fopen("m_cifrado","r");
                i=0;
                while((c=getc(mensaje))!=EOF)
                {
                        msg[i]=c;
                        i++;
                }
                msg[i+1]='\0';
                fclose(mensaje);
                remove("m_cifrado");
                if((he=gethostbyname(argv[1]))==NULL)
                {
                        printf("gethostbyname() error\n");
                        exit(-1);
                }
                if((fd=socket(AF_INET, SOCK_STREAM, 0))==-1)
                {
                        printf("socket() error\n");
                        exit(-1);
                }
                server.sin_family = AF_INET;
                server.sin_port = htons(PORT);
                server.sin_addr = *((struct in_addr *)he->h_addr);
                bzero(&(server.sin_zero),8);
                if(connect(fd, (struct sockaddr *)&server,sizeof(struct sockaddr))==-1)
                {
                        printf("connect() error\n");
                        exit(-1);
                }
                len=strlen(msg);
                if((bytes_sent=send(fd,msg,len+1,0)) == -1)
                {
                        printf("Error en send() \n");
                        exit(-1);
                }
                if((numbytes=recv(fd,buf,MAXDATASIZE,0)) == -1)
                {
                        printf("Error en recv() \n");
                        exit(-1);
                }
                buf[numbytes]='\0';
                close(fd);

        execlp(INSTALL_PATH_CASTOR"/bin/service_down_server","service_down_server",NULL
);
        }
        execlp(INSTALL_PATH_CASTOR"/bin/crypt","crypt","switchover",CLAVE,NULL);
        return 0;
}
```

5.4.3.2.1.11 crypt

Este módulo cifra la comunicación entre los Nodos del *cluster*, garantizando la confidencialidad del medio.

```
/*################################################*/
/*PROGRAMA QUE CIFRA EL MENSAJE A ENVIAR A NODO 2*/
/*################################################*/

#include <stdio.h>
#include "dioscuri_config.h"

main(argc,argv)
int argc;
char *argv[];
{

 FILE *fp;
 int pfds[2],i,len;
 char caracter[1],cadena[100];
 if(argc==3)
 {
        strcpy(cadena,argv[1]);
        len=strlen(cadena);
        fp=fopen("m_claro","w");
        for(i=0;i<len;i++)
                putc(cadena[i],fp);
        fclose(fp);
        pipe(pfds);
        if(!fork())
        {
                close(1);
                dup(pfds[1]);
                close(pfds[0]);
                execlp(OPENSSL,"openssl","des3","-nosalt","-in","m_claro","-k",argv[2],(char *)0);
        }
        else
        {
                close(0);
                dup(pfds[0]);
                close(pfds[1]);
                fp=fopen("m_cifrado","w");
                while(read(pfds[0],caracter,1))
                fwrite(caracter,1,1,fp);
                fclose(fp);
                remove("m_claro");
        }
 }
 else
        printf("ESCRIBE LA PALABRA A CIFRAR Y LA CLAVE\n");
}
```

5.4.3.2.1.12 service_down_server

Este módulo ejecuta la detención del Grupo de Servicios en el Nodo.

```
/*#####################################################################*/
/*PROGRAMA QUE EJECUTA LA DETENCION DE LOS SERVICIOS DEL WEB SERVER*/
/*#####################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <errno.h>
#include <syslog.h>
#include "dioscuri_config.h"

int main(int argc, char *argv[])
{
        int demonio,demonio2;
        demonio=fork();
        if(demonio<0)
        {
                perror(errno);
                return -1;
        }
        else if(demonio!=0)
        {
                wait();
                demonio2=fork();
                if(demonio2<0)
                {
                        perror(errno);
                return -1;
                }
                else if(demonio2!=0)
                {
                        wait();
                        openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);

                        syslog(LOG_EMERG,"El Web Server esta fuera de actividad.");
                }

        execlp(INSTALL_PATH_CASTOR"/bin/service_down_server2","service_down_server2",NU
LL);
                return 0;
        }
        execlp(NETWORK,"network","stop",NULL);
        return 0;
}
```

5.4.3.2.1.13 service_down_server2

Propiamente, este módulo realiza la detención del Grupo de Servicios.

```
#####################################################################
#PROGRAMA QUE REALIZA LA DETENCION DE LOS SERVICIOS DEL NODO 1
#####################################################################


/etc/init.d/sshd stop >> /dev/null
/usr/local/apache/bin/apachectl stop >> /dev/null
killall postmaster >> /dev/null
killall monitor_server >> /dev/null
```

5.4.3.2.1.14 castor.conf

Este archivo de configuración secundario, se encuentra ubicado en el directorio "conf", el funcionamiento de algunos módulos depende de su existencia y correcta configuración.

```
#############################################################################
#ARCHIVO DE CONFIGURACION SECUNDARIO DEL SISTEMA DE ALTA DISPONIBILIDAD
#EN EL NODO CASTOR
#############################################################################

#VARIABLES DE ENTORNO
INSTALL_PATH_CASTOR="/root/ADSWL/dioscuri-1.0/castor"

#DATOS DE LOS NODOS Y EL GATEWAY
IP_NODO1="192.168.0.1"
IP_NODO2="192.168.0.2"
GATEWAY="191.168.0.254"
HOSTNAME_NODO1="host1.fi-a.unam.mx"
HOSTNAME_NODO2="host2.fi-a.unam.mx"

#TIEMPO EN EL CUAL SE COMPROBARA LA CONECTIVIDAD DE LOS NODOS
PERIODO_HEARTBEAT=10

#VARIABLES EMPLEADAS EN LA SINCRONIZACION DE LA INFORMACION

#COMANDO RSYNC
RSYNC="/usr/bin/rsync"

#DIRECTORIOS DE LOS USUARIOS A RESPALDAR
USERS_DIR="/casa /usr/local/pgsql/data /home /usuarios /users"
```

5.4.3.2.2 CÓDIGO FUENTE DE POLLUX

POLLUX está compuesto por los siguientes módulos (Ver 5.12):

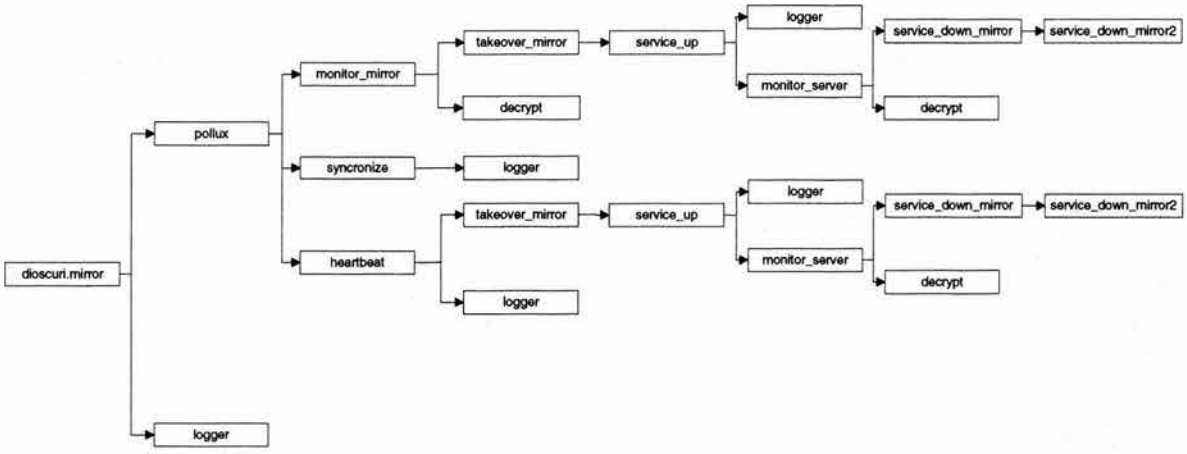

Figura 5.12 Módulos de POLLUX

5.4.3.2.2.1 dioscuri.mirror

Este módulo tiene la función de ser el menú principal de operaciones, recibe los parámetros *start, y halting*. La Tabla 5.14 ilustra las acciones que realiza con cada parámetro.

PARÁMETRO	ACCIÓN BÁSICA
start	Ejecuta el módulo *pollux*
halting	Ejecuta una rutina que realiza la detención del sistema, ésta rutina viene incluida en el módulo

Tabla 5.14 Parámetros de funcionamiento

El código fuente de *dioscuri.mirror* es el siguiente:

```
#####################################################
#MENU PRINCIPAL DE OPERACIONES PARA POLLUX
#INICIA O DETIENE EL SISTEMA DE ALTA DISPONIBILIDAD
#####################################################
. ../conf/pollux.conf

case $1 in

start)          clear
                echo ""
                if [ -f "$INSTALL_PATH_POLLUX""/bin/pollux" ]
                then
                echo -n  "Iniciando el Sistema de Alta Disponibilidad...  "
                "$INSTALL_PATH_POLLUX""/bin/pollux"&
                "$INSTALL_PATH_POLLUX""/bin/logger" "El Sistema de Alta Disponibilidad ha
sido activado exitosamente."
```

```
                echo "HECHO"
                        echo ""
                else
                echo "El Sistema de Alta Disponibilidad no puede ser iniciado."
                echo " El comando \"pollux\" no fue encontrado."
                "$INSTALL_PATH_POLLUX""/bin/logger" "El Sistema de Alta Disponibilidad no ha
sido activado debido a que el comando \"pollux\" no fue encontrado."
                        echo ""
                fi
                ;;
halting) clear
                echo "Ha decidido detener el Sistema de Alta Disponibilidad."
                echo ""
                echo
"*******************************************************************************"
                echo "ADVERTENCIA: Una vez hecho esto, detenga el Sistema de Alta
Disponibilidad en el Nodo 1."
                echo
"*******************************************************************************"
                echo ""
                echo -n "Proseguir con la detencion? [S/N] "
                read decision
                echo ""
                if [ $decision ]
                then
                        if [ $decision == S ]
                        then
                                echo "Ingrese el motivo por el cual se detendra el Sistema HA: "
                                echo ""
                                read motivo
                                if [ `echo $motivo|wc -c` -ne 1 ]
                                then
                                        killall monitor_server >> /dev/null
                                        cmd=$?
                                        clear
                                        if [ $cmd -ne 0 ]
                                        then
                                                killall monitor_mirror >> /dev/null
                                                killall syncronize >> /dev/null
                                                heartbeat=`cat
"$INSTALL_PATH_POLLUX""/heartbeat.pid"`
                                                chattr -i
"$INSTALL_PATH_POLLUX""/heartbeat.pid"
                                                rm -f "$INSTALL_PATH_POLLUX""/heartbeat.pid"
                                                kill -9 $heartbeat >> /dev/null
                                                syncronize=`cat
"$INSTALL_PATH_POLLUX""/syncronize.pid"`
                                                chattr -i
"$INSTALL_PATH_POLLUX""/syncronize.pid"
                                                rm -f "$INSTALL_PATH_POLLUX""/syncronize.pid"
                                                kill -9 $syncronize >> /dev/null
                                                cmd=$?
                                                clear
                                                echo ""
                                        echo -n "Parando el Sistema HA...   "
                                                if [ $cmd -eq 0 ]
```

```
                                        then
                                                echo "HECHO"
                                                echo ""
                                                "$INSTALL_PATH_POLLUX""/bin/logger"
"INTERRUPCION PLANEADA: $motivo"
                                                "$INSTALL_PATH_POLLUX""/bin/logger"
"El Sistema HA ha sido detenido."
                                        else
                                                echo "FALLO"
                                                echo ""
                                                echo "****************************************"
                                                echo "El Sistema HA no esta siendo
ejecutado."
                                                echo "****************************************"
                                                echo ""
                                                exit 0
                                        fi
                                else
                                        echo ""
                                echo -n "Parando el Sistema HA...  "
                                        if [ $cmd -eq 0 ]
                                then
                                        echo "HECHO"
                                        echo ""
                                        "$INSTALL_PATH_POLLUX""/bin/logger"
"INTERRUPCION PLANEADA: $motivo"
                                        "$INSTALL_PATH_POLLUX""/bin/logger" "El Sistema HA
ha sido detenido."
                                else
                                echo "FALLO"
                                        echo ""
                                        echo "****************************************"
                                        echo "El Sistema HA no esta siendo ejecutado."
                                        echo "****************************************"
                                        echo ""
                                        exit 0
                                fi
                                        fi
                        else
                                echo ""
                                echo "*************************************************************************"
                                echo "Debe especificar el motivo por le cual quiere usted detener el
Sistema HA."
                                echo "*************************************************************************"
                                echo ""
                                        fi
                                else
                                        echo ""
                                        echo "************************************************************"
                                        echo "El Sistema de Alta Disponibilidad no ha sido modificado."
                                        echo "************************************************************"
                                        echo ""
                                fi
                        else
                        echo ""
                       echo "************************************************************"
```

```
                echo "El Sistema de Alta Disponibilidad no ha sido modificado."
                echo "*********************************************************"
                echo ""
                 fi
                 ;;
*)               echo "PARAMETROS DE USO: [start|halting] "
                 exit 0
esac
```

5.4.3.2.2.2 pollux

Módulo principal que ejecuta los demonios *monitor_mirror, syncronize* y *hearbeat.*

```
#############################################################################
#PROGRAMA QUE LEVANTA LOS DEMONIOS MONITOR_MIRROR, SYNCRONIZE Y
HEARTBEAT
#############################################################################
. ../conf/pollux.conf

"$INSTALL_PATH_POLLUX""/bin/monitor_mirror"&

"$INSTALL_PATH_POLLUX""/bin/syncronize"&

echo $! > "$INSTALL_PATH_POLLUX""/syncronize.pid"
chattr +i "$INSTALL_PATH_POLLUX""/syncronize.pid"

"$INSTALL_PATH_POLLUX""/bin/heartbeat"&

echo $! > "$INSTALL_PATH_POLLUX""/heartbeat.pid"
chattr +i "$INSTALL_PATH_POLLUX""/heartbeat.pid"
```

5.4.3.2.2.3 logger

Módulo que registra la actividad del Sistema de Alta Disponibilidad.

```
/*#############################################################################
###*/
/*PROGRAMA QUE GENERA UNA ENTRADA EN LA BITACORA DEL SISTEMA DE ALTA
DISPONIBILIDAD*/
/*#############################################################################
###*/
#include <stdio.h>
#include <syslog.h>
main(argc,argv)
int argc;
char *argv[];
{
        if (argc !=2)
        {
                printf("Uso: %s REGISTRO\n",argv[0]);
                exit(-1);
        }
        char mensaje[100];
        strcpy(mensaje,argv[1]);
```

```
    openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);
    syslog(LOG_EMERG,mensaje);
}
```

5.4.3.2.2.4 monitor_mirror

Este módulo abre un canal de comunicación con el Nodo 1 para saber si uno o varios servicios han fallado, en este caso, realiza la toma del Grupo de Servicios.

```
/*#############################################################################*/
/*PROGRAMA QUE ESCUCHA EL FALLO EN EL GRUPO DE SERVICIOS DEL NODO 1, EN SU
CASO*/
/*REALIZA TAKEOVER SUPLANTANDO A NODO 1 Y APODERANDOSE DEL GRUPO DE
SERVICIOS */
/*#############################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <time.h>
#include "dioscuri_config.h"

main()
{
        FILE *temporal;
        int fd, fd2,numbytes,i,len,demonio,demonio2;
        char buf[MAXDATASIZE];
        struct sockaddr_in server;
        struct sockaddr_in client;
        int sin_size;
        demonio=fork();
        if(demonio<0)
        {
        perror(errno);
        return -1;
        }
        else if(demonio != 0)
        {
                wait();
                demonio2=fork();
                if(demonio2<0)
                {
                        perror(errno);
                        return -1;
                }
                else if(demonio2!=0)
                {
                        wait();
                retardo(10);

        execlp(INSTALL_PATH_POLLUX"/bin/takeover_mirror","takeover_mirror",NULL);
                        exit(-1);
                }
```

```
                execlp(INSTALL_PATH_POLLUX"/bin/decrypt","decrypt",CLAVE,NULL);
                return 0;
                exit(-1);
        }
        if ((fd=socket(AF_INET, SOCK_STREAM, 0)) == -1 )
        {
                printf("error en socket()\n");
                exit(-1);
        }
        server.sin_family = AF_INET;
        server.sin_port = htons(PORT);
        server.sin_addr.s_addr = INADDR_ANY;
        bzero(&(server.sin_zero),8);
        if(bind(fd,(struct sockaddr*)&server,sizeof(struct sockaddr))==-1)
        {
                printf("error en bind() \n");
                exit(-1);
        }
        if(listen(fd,BACKLOG) == -1)
        {
                printf("error en listen()\n");
                exit(-1);
        }
        while(1)
        {
                sin_size=sizeof(struct sockaddr_in);
                if ((fd2 = accept(fd,(struct sockaddr *)&client,&sin_size))==-1)          {
                        printf("error en accept()\n");
                        exit(-1);
                }
                if ((numbytes=recv(fd2,buf,MAXDATASIZE,0)) == -1)
                {
                        printf("Error en recv() \n");
                        exit(-1);
                }
                buf[numbytes]='\0';
        len=strlen(buf);
        temporal=fopen("m_cifrado","w");
        for(i=0;i<len;i++)
                putc(buf[i],temporal);
                putc(EOF,temporal);
        fclose(temporal);
                close(fd2);
                return 0;
        }
}
```

5.4.3.2.2.5 syncronize

Este módulo sincroniza la información existente en el Nodo 1 de forma periódica.

```
#########################################################################################
#######################
#PROGRAMA QUE SINCRONIZA LA INFORMACION DE LOS USUARIOS DEL NODO 1 Y
REPORTA EN BITACORA DICHO EVENTO
```

```
#############################################################################
#####################
. ../conf/pollux.conf

while [ 1 ]
do
        HEARTBEAT_NODO2=`ping -w 1 -c 1 $GATEWAY 2> /dev/null | grep packets | awk '{print
$6}'`
        if [ $HEARTBEAT_NODO2 == 0% ]
        then
                HEARTBEAT_NODO1=`ping -w 1 -c 1 $IP_NODO1 2> /dev/null | grep packets |
awk '{print $6}'`
                if [ $HEARTBEAT_NODO1 == 0% ]
                then
                        for i in $USERS_DIR
                        do
                                $RSYNC -ave ssh $IP_NODO1:$i/* $i/  >> /dev/null
                                "$INSTALL_PATH_POLLUX""/bin/logger" "la informacion del
directorio $i esta sincronizada."
                        done
                        rm -f /usr/local/pgsql/data/postmaster.pid
                else
                        "$INSTALL_PATH_POLLUX""/bin/logger" "Fallo de conectividad del Web
Server (Nodo 1) al tratar de sincronizar la informacion."
                fi
        else
                "$INSTALL_PATH_POLLUX""/bin/logger" "Fallo de conectividad del Mirror Server
(Nodo 2) al tratar de sincronizar la informacion."
        fi
        sleep $PERIODO_SYNCRONIZE
done
```

5.4.3.2.2.6 hearbeat

Este módulo realiza una comprobación de conectividad del Nodo 1, como se dice en el argot, verifica la "salud" del Nodo, en caso de no responder, realize la toma del Grupo de Servicios.

```
#############################################################################
##############
#PROGRAMA QUE REALIZA LA COMPROBACION DE CONECTIVIDAD DEL NODO 1, SI ESTE
NO ESTA DISPONIBLE,
#EJECUTA TAKEOVER Y TOMA EL GRUPO DE SERVICIOS
#############################################################################
##############
. ../conf/pollux.conf

while [ 1 ]
do
    HEARTBEAT_NODO2=`ping -w 1 -c 1 $GATEWAY | grep packets | awk '{print $6}'`
    if [ $HEARTBEAT_NODO2 == "100%" ]
    then
        "$INSTALL_PATH_POLLUX""/bin/logger" "Falla de conectividad del Mirror Server (Nodo
2)."
```

```
    else
        HEARTBEAT_NODO1=`ping -w 1 -c 1 $IP_NODO1 | grep packets | awk '{print $6}'`
        if [ $HEARTBEAT_NODO1 == "100%" ]
        then
            "$INSTALL_PATH_POLLUX""/bin/logger" "Falla de conectividad del Web Server
(Nodo 1)."
            break
        fi
    fi
    sleep $PERIODO_HEARTBEAT
done

killall monitor_mirror >> /dev/null
"$INSTALL_PATH_POLLUX""/bin/takeover_mirror"&

chattr -i "$INSTALL_PATH_POLLUX""/heartbeat.pid"
rm -f "$INSTALL_PATH_POLLUX""/heartbeat.pid"

chattr -i "$INSTALL_PATH_POLLUX""/syncronize.pid"
rm -f "$INSTALL_PATH_POLLUX""/syncronize.pid"
```

5.4.3.2.2.7 takeover_mirror

takeover_mirror realiza propiamente la toma del Grupo de Servicios del Nodo caido.

```
/*#####################################################################*/
/*PROGRAMA QUE REALIZA EL TAKEOVER DEL GRUPO DE SERVICIOS DEL NODO 1*/
/*#####################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <syslog.h>
#include "dioscuri_config.h"

main()
{
int demonio,demonio2,demonio3;
demonio=fork();
if(demonio<0)
{
    perror(errno);
    return -1;
}
else if(demonio!=0)
   {
            wait();
            demonio2=fork();
            if(demonio2<0)
            {
                perror(errno);
                return -1;
```

```
                }
                else if(demonio2!=0)
                   {
                                wait();
                        openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);
                        syslog(LOG_EMERG,"El Mirror Server (Nodo 2) esta activo. Ha suplantado
a Nodo 1.");
                       demonio3=fork();
                     if(demonio3<0)
                {
                        perror(errno);
                        return -1;
                }
                else if(demonio3!=0)
                    {
                         wait();
                                        exit(-1);

                                }
                    execlp(INSTALL_PATH_POLLUX"/bin/service_up","service_up",NULL);
                                return 0;
                   }
                execlp(HOSTNAME,"hostname",HOSTNAME_NODO1,NULL);
                return 0;
   }

execlp(IFCONFIG,"ifconfig","eth0",IP_NODO1,"netmask",NETMASK,"broadcast",BROADCAST,"up
",NULL);
    return 0;
}
```

5.4.3.2.2.8 decrypt

Módulo que decifra los mensajes provenientes del Nodo 1.

```
/*#######################################################*/
/*PROGRAMA DE DESCIFRA EL MENSAJE PROVENIENTE DE NODO 1*/
/*#######################################################*/

#include <stdio.h>
#include "dioscuri_config.h"

main(argc,argv)
int argc;
char *argv[];
{
        FILE *fp;
        int pfds[2],i,len;
        char caracter[1],cadena[100];
        if(argc==2)
        {
                pipe(pfds);
                if(!fork())
                {
                        close(1);
```

```
                        dup(pfds[1]);
                        close(pfds[0]);
                        execlp(OPENSSL,"openssl","des3","-d","-nosalt","-in","m_cifrado","-
k",argv[1],(char *)0);
                }
                else
                {
                        close(0);
                        dup(pfds[0]);
                        close(pfds[1]);
                        fp=fopen("m_claro","w");
                        while(read(pfds[0],caracter,1))
                                fwrite(caracter,1,1,fp);
                        fclose(fp);
                        remove("m_cifrado");
                }
        }
        else
                printf("ESCRIBE LA PALABRA A CIFRAR Y LA CLAVE\n");
}
```

5.4.3.2.2.9 service_up

Módulo que inicia los demonios necesarios para que el Nodo 2 tome finalmente el Grupo de Servicios.

```
#################################################################
#PROGRAMA DE EJECUTA LOS DEMONIOS DE LOS SERVICIOS DEL NODO 2
#################################################################
. ../conf/pollux.conf

killall syncronize >> /dev/null

if ( test -f "$INSTALL_PATH_POLLUX""/heartbeat.pid" )
then
        heartbeat=`cat "$INSTALL_PATH_POLLUX""/heartbeat.pid"`
      chattr -i "$INSTALL_PATH_POLLUX""/heartbeat.pid"
      rm -f "$INSTALL_PATH_POLLUX""/heartbeat.pid"
      kill -9 $heartbeat >> /dev/null
fi

if ( test -f "$INSTALL_PATH_POLLUX""/syncronize.pid" )
then
      syncronize=`cat "$INSTALL_PATH_POLLUX""/syncronize.pid"`
      chattr -i "$INSTALL_PATH_POLLUX""/syncronize.pid"
      rm -f "$INSTALL_PATH_POLLUX""/syncronize.pid"
      kill -9 $syncronize >> /dev/null
fi

while [ 1 ]
do
      HEARTBEAT_NODO1=`ping -w 1 -c 1 $GATEWAY | grep packets | awk '{print $6}'`
      if [ $HEARTBEAT_NODO1 == "100%" ]
      then
```

```
            "$INSTALL_PATH_POLLUX""/bin/logger" "Falla de conectividad del Mirror Server (Nodo
2) despues de suplantar a Nodo 1."
            sleep $PERIODO_HEARTBEAT
      else
            /etc/init.d/sshd stop >> /dev/null
            /etc/init.d/sshd start >> /dev/null
            /usr/local/apache/bin/apachectl stop >> /dev/null
            /usr/local/apache/bin/apachectl start >> /dev/null
            killall postmaster >> /dev/null
            su -c '/usr/local/pgsql/bin/postmaster -i -S -D /usr/local/pgsql/data 2>&1 >
/var/log/postgresql.log' postgres& >> /dev/null
                  sleep 60
           "$INSTALL_PATH_POLLUX""/bin/monitor_server"&

            break
      fi
done
```

5.4.3.2.2.10 monitor_server

Este módulo abre un canal de comunicación en espera de que el Nodo 1 notifique que desea retomar el Grupo de Servicios. Ante esto, el Nodo 2 cede el Grupo.

```
/*##############################################################################
*/
/*PROGRAMA QUE ESTA A LA ESCUCHA SI NODO 1 DECIDE RETOMAR EL GRUPO DE
SERVICIOS */
/*Y EJECUTA UN SWITCHOVER QUE CEDE EL GRUPO                                    */
/*##############################################################################
*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <time.h>
#include "dioscuri_config.h"

main()
{
        FILE *temporal;
        int fd, fd2,numbytes,i,len,demonio,demonio2;
        char buf[MAXDATASIZE];
        struct sockaddr_in server;
        struct sockaddr_in client;
        int sin_size;
        demonio=fork();
        if(demonio<0)
        {
        perror(errno);
        return -1;
        }
        else if(demonio!=0)
        {
```

```
                wait();
                demonio2=fork();
                if(demonio2<0)
                {
                        perror(errno);
                        return -1;
                }
                else if(demonio2!=0)
                {
                        wait();
                        remove("m_claro");

        execlp(INSTALL_PATH_POLLUX"/bin/service_down_mirror","service_down_mirror",NULL);
                        exit(-1);
                }
                execlp(INSTALL_PATH_POLLUX"/bin/decrypt","decrypt",CLAVE,NULL);
}
if ((fd=socket(AF_INET, SOCK_STREAM, 0)) == -1 )
{
                printf("error en socket()\n");
                exit(-1);
}
server.sin_family = AF_INET;
server.sin_port = htons(PORT);
server.sin_addr.s_addr = INADDR_ANY;
bzero(&(server.sin_zero),8);
if(bind(fd,(struct sockaddr*)&server,sizeof(struct sockaddr))==-1)
{
                printf("error en bind() \n");
                exit(-1);
}
if(listen(fd,BACKLOG) == -1)
{
                printf("error en listen()\n");
                exit(-1);
}
while(1)
{
                sin_size=sizeof(struct sockaddr_in);
                if ((fd2 = accept(fd,(struct sockaddr *)&client,&sin_size))==-1)                {
                        printf("error en accept()\n");
                        exit(-1);
                }
                if ((numbytes=recv(fd2,buf,MAXDATASIZE,0)) == -1)
                {
                        printf("Error en recv() \n");
                        exit(-1);
                }
                buf[numbytes]='\0';
        len=strlen(buf);
        temporal=fopen("m_cifrado","w");
        for(i=0;i<len;i++)
                putc(buf[i],temporal);
                putc(EOF,temporal);
        fclose(temporal);
                close(fd2);
```

```
                return 0;
    }
}
```

5.4.3.2.2.11 service_down_mirror

Módulo que tiene la función de retomar la IP y el Hostname originales del Nodo 2. Ejecuta la detención del Grupo de Servcios en el Nodo.

```
/*#####################################################################*/
/*PROGRAMA QUE REALIZA LA DETENCION DEL SERVICIO DE RED DEL NODO 2 Y*/
/*LA RECONFIGURA DE TAL FORMA QUE NODO 2 RETOMA SU IP ORIGINAL          */
/*#####################################################################*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <errno.h>
#include <syslog.h>
#include "dioscuri_config.h"

int main(int argc, char *argv[])
{
        int demonio,demonio2,demonio3;
        demonio=fork();
        if(demonio<0)
        {
                perror(errno);
                return -1;
        }
        else if(demonio != 0)
        {
                wait();
                demonio2=fork();
                if(demonio2<0)
                {
                        perror(errno);
                        return -1;
                }
                else if(demonio2!=0)
                {
                        wait();
                        demonio3=fork();
                        if(demonio3<0)
                        {
                                perror(errno);
                                return -1;
                        }
                        else if(demonio3!=0)
                                {
                                        wait();
```

```
        execlp(INSTALL_PATH_POLLUX"/bin/service_down_mirror2","service_down_mirror2",NUL
L);
                                        exit(-1);
                                }

        openlog("dioscurid",LOG_CONS|LOG_PID|LOG_NDELAY,LOG_LOCAL1);
                                syslog(LOG_EMERG,"El Mirror Server ha cedido el Grupo de
Servicios al Web Server (Nodo 1).");
                                return 0;
                        }
          execlp(HOSTNAME,"hostname",HOSTNAME_NODO2,NULL);
           return 0;
        }
        execlp(IFCONFIG,"ifconfig","eth0",IP_NODO2,"netmask",NETMASK,"broadcast",BROADC
AST,"up",NULL);
      return 0;
}
```

5.4.3.2.2.12 service_down_mirror2

Reliza la detención del Grupo de Servicios y reinicializa el Sistema de Alta Disponibilidad a su estado inicial.

```
#############################################################################
#
#PROGRAMA QUE REALIZA LA DETENCION DE LOS SERVICIOS DEL NODO 2 Y
AUTOMATICAMENTE
#EJECUTA EL DEMONIO POLLUX GARANTIZANDO EL CONTINUO FUNCIONAMIENTO DEL
SISTEMA
#DE ALTA DISPONIBILIDAD
#############################################################################
#
. ../conf/pollux.conf

sleep 60
/etc/init.d/sshd stop >> /dev/null
/usr/local/apache/bin/apachectl stop >> /dev/null
killall postmaster >> /dev/null
"$INSTALL_PATH_POLLUX""/bin/dioscuri.mirror" start &
```

5.4.3.2.2.13 pollux.conf

Se encuentra ubicado en "*conf*", y para el buen funcionamiento de algunos módulos, debe existir y estar correctamente configurado.

```
#########################################################################
#ARCHIVO DE CONFIGURACION SECUNDARIO DEL SISTEMA DE ALTA DISPONIBILIDAD
#EN EL NODO POLLUX
#########################################################################

#VARIABLES DE ENTORNO
```

```
INSTALL_PATH_POLLUX="/root/ADSWL_2/pollux"


#DATOS DE LOS NODOS Y EL GATEWAY
IP_NODO2="192.168.0.2"
IP_NODO1="192.168.0.1"
GATEWAY="192.168.0.254"

#TIEMPO EN EL CUAL SE COMPROBARA LA CONECTIVIDAD DE LOS NODOS
PERIODO_HEARTBEAT=10

#VARIABLES EMPLEADAS EN LA SINCRONIZACION DE LA INFORMACION

#COMANDO RSYNC
RSYNC="/usr/local/bin/rsync"

#DIRECTORIOS DE LOS USUARIOS A RESPALDAR
USERS_DIR="/home /usuarios /users"

#TIEMPO ENTRE CADA SINCRONIZACION
PERIODO_SYNCRONIZE=60
```

5.4.3.2.3 EL ARCHIVO DE CONFIGURACIÓN PRIMARIO: dioscuri_config.h

Este archivo se encuentra dentro del directorio "*src*", y es un factor crítico de éxito el que esté correctamente configurado, previo a la instalación de la herramienta.

El codigo fuente es el siguiente:

```
/*#########################################################################*/
/*ARCHIVO DE CONFIGURACION PRIMARIO DEL SISTEMA DE ALTA DISPONIBILIDAD*/
/*NOTA: LAS VARIABLES DEFINIDAS EN ESTE ARCHIVO DEBEN SER COINCIDENTES*/
/*CON LAS QUE SE DEFINEN EN LOS ARCHIVOS DE CONFIGURACION SECUNDARIOS */
/*DE CADA NODO: pollux.conf y castor.conf                              */
/*#########################################################################*/

/*CONFIGURACION DE RED DE LOS NODOS DEL CLUSTER*/
#define IP_NODO1 "192.168.0.1"
#define IP_NODO2 "192.168.0.2"
#define HOSTNAME_NODO1 "host1.fi-a.unam.mx"
#define HOSTNAME_NODO2 "host2.fi-a.unam.mx"
#define NETMASK "255.255.255.0"
#define BROADCAST "192.168.0.255"

/*PUERTOS-SERVICIOS A MONITOREAR*/
#define PUERTO_SSH 22
#define PUERTO_HTTP 80
#define PUERTO_PGSQL 5432

/*CONSTANTES INVOLUCRADAS EN LA COMUNICACION ENTRE NODOS, DEFINE EL
PUERTO DE COMUNICACION*/
#define BACKLOG 2
#define PORT 3550
```

```
#define MAXDATASIZE 100

/*VARIABLES DE ENTORNO*/
#define NETWORK "/etc/init.d/network"
#define HOSTNAME "/bin/hostname"
#define IFCONFIG "/sbin/ifconfig"
#define OPENSSL "/usr/bin/openssl"
#define INSTALL_PATH_CASTOR "/root/ADSWL/dioscuri-1.0/castor"
#define INSTALL_PATH_POLLUX "/root/ADSWL_2/pollux"

/*CLAVE PARA EL CIFRADO DE LA COMUNICACION ENTRE NODOS*/
#define CLAVE "dioscuri"

/*DEFINICION DE LAS FUNCIONES PROTOTIPO GENERALES*/
void retardo(int);

/*DEFINICION DE FUNCIONES*/
void retardo(int segundos)
{
      int inicio;
      for(inicio=time(NULL)+segundos;time(NULL)!=inicio;time(NULL))
      {
      }
}
```

5.4.4 PROCESO DE INSTALACIÓN

Esta herramienta viene en un solo paquete llamado *dioscuri-1.0.tar.gz*, el cual primero habrá que descomprimirlo y desempaquetarlo.

```
[root@tajin root]# tar -zxvf dioscuri-1.0.tar.gz
```

Se creará un directorio llamado dioscuri-1.0, entramos en él:

```
[root@tajin root]# cd dioscuri-1.0/
[root@tajin root]# ls
castor
pollux
```

"*castor y pollux*" son dos directorios que contienen el código fuente necesario para construir el Sistema de Alta Disponibilidad. Estos directorios tiene el siguiente contenido:

```
[root@tajin root]# ls castor/
INSTALL
src
bin
conf
castor.install
```

El archivo "*INSTALL*" contiene las instrucciones para la instalación y configuración del Sistema de Alta Disponibilidad en CASTOR, el directorio "*src*" contiene el código fuente, y directorio "bin" contendrá una instancia de los binarios del sistema y el directorio "*conf*" contiene el archivo "*castor.conf*". Por último "*castor.install*" es un script que instala el Sistema de Alta Disponibilidad. El directorio "*pollux*"

contiene la misma estructura de archivos y directorios, sólo varía en el hecho de la existencia de "*pollux.install*" y "*pollux.conf*". La instalación de los Nodos es muy similar, sólo varia en la ejecución del script de instalación. Instalaremos el Nodo 1.

El contenido del archivo castor.install es el siguiente:

```
#! /bin/bash
###############################################
#SCRIPT DE INSTALACION DE CASTOR
###############################################
clear
echo ""
echo "Instalando el Sistema de Alta Disponibilidad en el Nodo 1..."
echo ""
echo -n ¿Cual sera el directorio de instalacion para CASTOR?
read dir_install
cd ./src
gcc –o ../bin/monitor_server monitor_server.c
gcc -o ../bin/logger logger.c
gcc -o ../bin/mantenance mantenance.c
gcc -o ../bin/takeover_server takeover_server.c
gcc -o ../bin/recover_ip recover_ip.c
gcc -o ../bin/switchover switchover.c
gcc -o ../bin/crypt crypt.c
gcc -o ../bin/service_down_server service_down_server.c
cp -f dioscuri.server ../bin
cp -f castor ../bin
cp -f restore_server ../bin
cp -f service_down_server2 ../bin
cp -f syncronize ../bin
cd ../bin
chmod 700 *
cd ../conf
chmod 700 castor.conf
cd ..
mkdir -p $dir_install
cp -rf bin conf $dir_install
echo ""
echo "El Sistema de Alta Disponibilidad en CASTOR ha sido instalado en $dir_install."
echo ""
echo "Para cualquier actualizacion debera ejecutar de nuevo este script."
echo ""
```

Antes de ejecutarlo, habrá que editar los archivos "*dioscuri_config.h*" (ubicado en el directorio "*src*" y "*castor.conf*" (ubicado en el directorio "*conf*"), de acuerdo a la configuración del servidor, deberá adaptarse. Hay que recordar que las variables definidas tanto en un archivo con en el otro deberán de concordar plenamente.

Ejecutaremos el script de esta forma:

```
[root@tajin root]# ./castor.install
```

E instalará el *Software* de Alta Disponibilidad en el Nodo 1.

Una vez hecho esto el siguiente paso es la configuración, para ello se realizará lo siguiente:

Agregar al archivo */etc/services* la siguiente línea:

```
dioscuri        6550        #dioscuri, Sistema HA
```

Donde 6550 es el puerto de comunicación entre los nodos del *cluster.*

Se creará un mecanismo de confianza mediante SSH para que *root* de Nodo 1 pueda conectarse directamente a Nodo 2 y viceversa. El comando para ello es *ssh-keygen*, (ver las páginas del manual en línea sobre *ssh* y *ssh-keygen*). Esto es importante para que la sincronización de la información sea automática, de lo contrario capa vez que se realice este proceso el sistema pedirá el *password* de root. Mediante IPTABLES, bloquearemos el acceso a todos por medio del puerto especificado en el archivo */etc/services* y permitiremos el acceso sólo al Nodo 2.

El Sistema de Alta Disponibilidad se apoya de dos herramientas de seguridad: OpenSSL y RSYNC, ambas deberán de estar presentes en el sistema.

Para iniciar el Sistema de Alta Disponibilidad, emplearemos el comando *dioscuri.server* ubicado dentro del directorio "*bin*" de la ruta especificada durante la instalación. Supongamos que el lugar de la instalación fue */usr/local/dioscuri/castor*, entonces para iniciar el Sistema de Alta Disponibilidad teclearemos:

```
[root@tajin root]# /usr/local/dioscuri/castor/bin/dioscuri.server start
```

El proceso de instalación, configuración y los requerimientos en cuanto a *software* son los mismos para el Nodo 2, y para ejecutar el sistema, existe también un comando llamado "*dioscuri.mirror*", ubicado en el directorio de instalación de POLLUX.

CAPÍTULO 6

Implementación del Sistema de Alta Disponibilidad

CAPÍTULO 6. IMPLEMENTACIÓN DEL SISTEMA DE ALTA DISPONIBILIDAD

Una vez que se han superado las fases de desarrollo y pruebas del Sistema de Alta Disponibilidad, la siguiente tarea será la implementación en un ambiente de producción. Esta fase involucra aspectos tanto técnicos como administrativos. En el aspecto técnico podemos hablar propiamente de la eliminación de Puntos Simples de Fallo con los cuales se trabajaron en el análisis, diseño y desarrollo del Sistema de Alta Disponibilidad, esto incluye evidentemente la instalación y configuración tanto del *Cluster* de Alta Disponibilidad como del mismo *software*.

Sin embargo, la implementación no acaba aquí, habrá que garantizar su continuidad y buen funcionamiento. Para ello habrá que definir ciertos procedimientos que serán necesarios como son: el fomentar la cultura del uso de un ambiente de desarrollo y pruebas, el mantenimiento de una caja de repuestos, la elaboración de un Plan de Contingencias y por último la capacitación para el grupo de administradores y operadores del Sistema de Alta Disponibilidad.

6.1 USO DE UN AMBIENTE DE DESARROLLO Y PRUEBAS

Las tareas de investigación y desarrollo son actividades que nunca descansan, por ello, es recomendable crear un ambiente independiente y aislado del sistema en producción que permita a los administradores continuar desarrollando y probando nuevas herramientas y configuraciones que mejoren las que actualmente se utilizan. Un ambiente de desarrollo y pruebas requiere de equipo destinado exclusivamente para este fin, recursos humanos con el perfil adecuado para desempeñar la tarea y programas de calidad y planeación que conduzcan adecuadamente la actividad y la lleven a producir resultados favorables.

6.2 MANTENIMIENTO DE UNA CAJA DE REPUESTOS

Supongamos un escenario en el cual nuestro Sistema de Alta Disponibilidad sufre un daño en el *hardware* del equipo, por ejemplo en la fuente de poder de uno de los nodos. Si este Punto Simple de Fallo no fue eliminado, instalando una fuente de poder redundante, entonces se produce una interrupción no planeada del servicio y su duración estará en función del tiempo que tarde el administrador en detectar el fallo, en desinstalar la fuente de poder dañada, en conseguir rápidamente otra fuente en buen estado, en instalar dicha fuente y por último en encender el equipo. Todos estos tiempos, pueden reducirse en función de la habilidad del administrador, sin embargo, el hecho de encontrar una fuente de poder está prácticamente en función de la disponibilidad de la misma en el lugar.

Por ello, es recomendable crear y mantener una caja de repuestos de *hardware*, la cual deberá contener en primera instancia, discos duros, fuentes de poder, tarjetas de expansión, tarjetas de red, algunos módulos de memoria RAM, cable de red, conectores, cables de corriente, entre otros.

Evidentemente el mantenimiento de una caja de repuestos debe ser llevado con suma responsabilidad, los repuestos deben ser nuevos o en buen estado pero lo más importante es que funcionen adecuadamente, puesto que estos repuestos serán utilizados en casos críticos.

El contar con una caja de repuestos a la mano, podrá minimizar el tiempo que tarde en ser atendida una interrupción no planeada del servicio.

6.3 ELABORACIÓN DE UN PLAN DE CONTINGENCIAS

Un Plan de Contingencias básicamente es la estrategia a seguir ante un desastre natural que atente contra la disponibilidad e integridad del sistema en producción.

Bajo la presión de una catástrofe, el contar con un procedimiento local o remoto puede ser la gran diferencia.

El Plan de Contingencias implica un análisis de los posibles riesgos a los cuales pueden estar expuestos los equipos de cómputo y la información contenida en los diversos medios de almacenamiento, un análisis del cómo reducir su posibilidad de ocurrencia y los procedimientos a seguir en caso que se presentara el problema. Pese a todas las medidas de seguridad puede ocurrir un desastre, por tanto es necesario que el Plan de Contingencias incluya un Plan de Recuperación de Desastres, el cual tendrá como objetivo, restaurar el servicio en forma rápida, eficiente y con el menor costo y pérdidas posibles.

Si bien es cierto que se pueden presentar diferentes niveles de daños, también se hace necesario presuponer que el daño ha sido total, con la finalidad de tener un Plan de Contingencias lo más completo posible. El Plan de Contingencias contendrá los siguientes aspectos:

1. Plan de Reducción de Riesgos
2. Plan de Recuperación de Desastres
    - 2.1. Actividades previas al desastre
        - 2.1.1. Establecimiento del Plan de Acción
        - 2.1.2. Formación de equipos operativos
        - 2.1.3. Formación de equipos de evaluación
    - 2.2. Actividades durante el desastre
        - 2.2.1. Plan de Emergencias
        - 2.2.2. Formación de equipos
        - 2.2.3. Entrenamiento
    - 2.3. Actividades después del desastre
        - 2.3.1. Evaluación de daños
        - 2.3.2. Priorización de actividades del Plan de Acción señaladas en el punto 2.1.1.
        - 2.3.3. Ejecución de Actividades
        - 2.3.4. Evaluación de Resultados
        - 2.3.5. Retroalimentación del Plan de Acción

6.3.1 ANÁLISIS DE RIESGOS

El Análisis de Riesgos supone más que el hecho de calcular la posibilidad de que ocurran cosas negativas. Se ha de poder obtener una evaluación económica del impacto de estos sucesos negativos. Este valor se podrá utilizar para contrastar el costo de la protección de la Información en análisis, contra el costo de volverla a producir.

La evaluación de riesgos y presentación de respuestas debe prepararse de forma personalizada para cada organización.

La evaluación de riesgos supone imaginarse lo que puede ir mal y a continuación estimar el costo que supondría. Se ha de tener en cuenta la probabilidad de que sucedan cada uno de los problemas posibles. De esta forma se pueden priorizar los problemas y su costo potencial desarrollando un plan de acción adecuado.

Para este caso en particular, los administradores del Sistema de Alta Disponibilidad podrán apoyarse en el Análisis de Riesgos o Esquema de Seguridad desarrollado en el Capítulo 2 para este fin.

6.3.2 PLAN DE RECUPERACIÓN DE DESASTRES

Es importante definir los procedimientos y planes de acción para el caso de una posible falla, siniestro o desastre. Cuando ocurra una contingencia, es esencial que se conozca al detalle el motivo que la originó y el daño producido, lo que permitirá recuperar en el menor tiempo posible el proceso perdido. La elaboración de los procedimientos que se determinen como adecuados para un caso de emergencia, deben ser planeados y probados fehacientemente.

Los procedimientos deberán ser de ejecución obligatoria y bajo la responsabilidad de los encargados de la realización de los mismos, debiendo haber procesos de verificación de su cumplimiento. En estos procedimientos estará involucrado todo el personal de la Institución.

Los procedimientos de planes de recuperación de desastres deben de emanar de la máxima autoridad Institucional, para garantizar su difusión y estricto cumplimiento.

Las actividades a realizar en un Plan de Recuperación de Desastres se pueden clasificar en tres etapas (Ver Tabla 6.1):

Actividades Previas al Desastre
Actividades Durante el Desastre
Actividades Después del Desastre

Tabla 6.1 Componentes de un Plan de Recuperación de Desastres

6.3.2.1 ACTIVIDADES PREVIAS AL DESASTRE

Son todas las actividades de planeación, preparación, entrenamiento y ejecución de las actividades de resguardo de la información, que nos aseguren un proceso de recuperación con el menor costo posible. A continuación se describen dichas actividades.

6.3.2.1.1 ESTABLECIMIENTO DEL PLAN DE ACCIÓN

En esta fase se deben definir los procedimientos relativos a:

Sistemas e Información: La institución deberá tener una relación de los Sistemas de Información con los que cuenta. La relación deberá detallar lo siguiente (Ver Tabla 6.2):

Nombre del sistema
Lenguaje o paquete con el que fue desarrollado
La dirección que genera la información del sistema
Los departamentos que usan la información del sistema
El volumen de archivos con los que trabaja el sistema
El volumen de transacciones diarias, semanales y mensuales que maneja el sistema
El equipo necesario para su manejo óptimo
Las fechas en las que la información es necesitada con carácter de urgencia
El nivel de importancia estratégica que tiene la información de este sistema para la institución
Actividades de restauración del sistema

Tabla 6.2 Detalles de la relación sobre Sistemas e Información

Con toda esta información se deberá realizar un listado jerárquico de los Sistemas de Información necesarios para que la institución pueda recuperar su operatividad perdida en el desastre.

Equipos de Cómputo: Tener en cuenta la siguiente información (Ver Tabla 6.3):

Inventario actualizado de los equipos de manejo de información
Pólizas de seguros comerciales
Etiquetado de los equipos de acuerdo a la importancia de su contenido
Relación actualizada del equipo requerido como mínimo para cada Sistema, las funciones que realizaría y su posible uso en dos o tres turnos de trabajo

Tabla 6.3 Información sobre el Equipo de Cómputo

Obtención y almacenamiento de los respaldos de información: Se deberá establecer los procedimientos para la obtención de Copias de Seguridad de todos los elementos de *software* necesarios para asegurar la correcta ejecución de los Sistemas. Por tanto se requiere lo siguiente (Ver Tabla 6.4):

Respaldos del sistema operativo
Respaldos del *software* base
Respaldos del *software* aplicativo o de los Sistema de Información
Respaldos de los datos
Respaldo del *hardware* critico

Tabla 6.4 Tipos de respaldos

Políticas (Normas y Procedimientos de respaldo): Se debe establecer los procedimientos, normas y determinación de responsabilidades en la obtención de los respaldos mencionados anteriormente, incluyendo lo siguiente (Ver Tabla 6.5):

Periodicidad de cada tipo de respaldo
Respaldo de información entre los periodos que no se generan respaldos completos (respaldos incrementales)
Uso obligatorio de un formulario para el registro y control de los respaldos
Correlación entre los Sistemas e Información necesarias para la buena marcha de la institución y los respaldos efectuados
Almacenamiento de los respaldos en condiciones ambientales optimas, dependiendo del medio magnético empleado
Reemplazo de los respaldos, en forma periódica, antes que el medio magnético de soporte se deteriore
Almacenamiento de los respaldos en lugares distintos donde reside la información original
Pruebas periódicas de los respaldos, verificando su funcionalidad

Tabla 6.5 Políticas

6.3.2.1.2 FORMACIÓN DE EQUIPOS OPERATIVOS

En cada unidad operativa que almacene información y sirva para la operatividad institucional, se deberá designar un responsable de la seguridad de la información de su unidad. Sus labores se describen en la siguiente Tabla 6.6:

Ponerse en contacto con los propietarios de las aplicaciones y trabajar en conjunto
Proporcionar soporte técnico para las copias de respaldo de las aplicaciones
Planificar y establecer los requerimientos de los sistemas operativos en cuanto a archivos, bibliotecas, utilidades, entre otros, para los principales sistemas
Supervisar procedimientos de respaldo y restauración
Supervisar la carga de archivos de datos de las aplicaciones y la creación de archivos incrementales
Coordinar líneas, terminales, *modems* y cualquier otro equipo de comunicación
Establecer procedimientos de seguridad en los sitios de recuperación
Organizar las pruebas de *hardware* y *software*
Ejecutar trabajos de recuperación
Cargar y probar archivos del sistema operativo y otros sistemas almacenados en un sitio alterno
Realizar procedimientos de control de inventario y seguridad del almacenamiento en el sitio alterno
Establecer y llevar a cabo procedimientos para restaurar el lugar de recuperación
Participar en las pruebas y simulacros de desastres

Tabla 6.6 Labores del responsable de los Equipos Operativos

6.3.2.1.3 FORMACIÓN DE EQUIPOS DE EVALUACIÓN

Las funciones, responsabilidades y objetivos de este equipo deberán estar claramente definidas. Entre ellas se deberán incluir las siguientes (Ver Tabla 6.7):

Revisar que las normas y procedimientos con respecto a los respaldos y la seguridad de los equipos se cumpla
Supervisar la realización periódica de los respaldos comprobando físicamente su realización, adecuado registro y almacenamiento
Revisar la correlación entre los Sistemas e Información necesarias para la buena marcha de la institución y los respaldos realizados
Informar de los cumplimientos e incumplimientos de las normas para las acciones de corrección respectivas

Tabla 6.7 Funciones básicas de los Equipos de Evaluación

6.3.2.2 ACTIVIDADES DURANTE EL DESASTRE

Una vez presentada la Contingencia, se deberá ejecutar las siguientes actividades planeadas previamente.

6.3.2.2.1 PLAN DE EMERGENCIAS

Se establecen las acciones que se deben realizar cuando se presente una contingencia, así como la difusión de las mismas. Es conveniente prever los posibles escenarios: durante el día, durante la noche o madrugada. Este Plan deberá incluir la participación y actividades a realizar por todas las personas que se pueden encontrar presentes durante la contingencia, debiendo detallar las vías de salida o escape, un Plan de Evacuación de Personal, un Plan de protección de los activos de la institución si las circunstancias lo permiten, ubicación y señalización de los elementos contra el siniestro, elementos de iluminación, lista de teléfonos de bomberos, ambulancias, seguridad, entre otros.

6.3.2.2.2 FORMACIÓN DE EQUIPOS

Definir claramente las funciones a ejecutar durante la contingencia por parte de esta agrupación, teniendo como objetivo principal la protección de la integridad del personal y en caso de que la situación lo permita deberán existir dos equipos, uno de ellos encargado de combatir el siniestro y otro para el salvamento de los recursos informáticos críticos.

6.3.2.2.3 ENTRENAMIENTO

Establecer un programa de practicas periódicas de todo el personal ante el posible suceso de una contingencia, y de acuerdo a los diferentes tipos de siniestros y los roles que se hayan asignado en los planes de evacuación de personal. Un aspecto importante es que el personal tome conciencia del hecho y se le otorgue la importancia debida, para esto es necesario la participación activa de la directiva.

6.3.2.3 ACTIVIDADES DESPUES DEL DESASTRE

Estas actividades se detallan a continuación:

6.3.2.3.1 EVALUACIÓN DE DAÑOS

Se deberá evaluar la magnitud del daño producido una vez que la contingencia ha finalizado, considerando los sistemas afectados, los equipos que ha quedado innoperatibles, los que se pueden recuperar, entre otros aspectos.

6.3.2.3.2 PRIORIZACIÓN DE LAS ACTIVIDADES DEL PLAN DE ACCIÓN

Puesto que el Plan de Acción es general y contempla la perdida total, su comparación entre este y el resultado generado por la evaluación de daños nos dará la lista de actividades que se deben realizar de manera prioritaria y estratégica.

6.3.2.3.3 EJECUCIÓN DE ACTIVIDADES

Implica la creación de equipos de trabajo previamente planificados en el Plan de Acción. Cada equipo contara con un coordinador que deberá reportar diariamente el avance de los trabajos de recuperación. Estos trabajos tendrán dos etapas, la primera es la restauración del servicio usando los recursos de sitio alterno o de respaldo, y la segunda es volver a contar con los recursos en las cantidades y lugares propios del Sistema de Información.

6.3.2.3.4 EVALUACIÓN DE RESULTADOS

Una vez concluidas las labores de recuperación de los sistemas que fueron afectados, se realizara una evaluación objetiva de ellas. De la evaluación de resultados y de la Contingencia propiamente, deberán salir dos tipos recomendaciones, una servirá para la retroalimentación del Plan de Contingencias y la otra servirá para minimizar los riesgos y pérdida que ocasionaron el siniestro.

6.3.2.3.5 RETROALIMENTACIÓN DEL PLAN DE ACCIÓN

A su vez, con la evaluación de resultados, se deberá optimizar el Plan de Acción original, mejorando las actividades que tuvieron algún tipo de dificultad y reforzando los elementos que funcionaron adecuadamente.

Otro aspecto importante es el evaluar el costo que hubiera generado la falta de un Plan de Contingencias para la institución.

6.4 CAPACITACIÓN PARA EL EQUIPO DE ADMINISTRADORES DEL SISTEMA DE ALTA DISPONIBILIDAD

El que una empresa o institución cuente con un Sistema de Alta Disponibilidad no garantiza que sus servicios estén 100% disponibles. La disponibilidad, por ser un servicio de seguridad, hereda las características de la seguridad en un sistema, una de estas características es que ningún sistema es 100% seguro. Por ello, parte fundamental de un Sistema de Alta Disponibilidad es el factor humano, el cual interviene tanto en la administración como en aquellos escenarios donde el resto del Sistema de Alta Disponibilidad no puede actuar, el grupo de administradores deberá estar plenamente capacitado para reconocer y tomar las acciones preventivas y correctivas necesarias.

La capacitación deberá ser impartida en términos de Alta Disponibilidad, es decir, que las acciones y procedimientos de los administradores tengan como objetivo fundamental el minimizar en lo posible las interrupciones planeadas y prevenir la aparición de las interrupciones no planeadas.

La capacitación deberá tocar fundamentalmente los siguientes aspectos (Ver Tabla 6.8):

Instalación y configuración del Cluster de Alta Disponibilidad
Administración del *software* de Alta Disponibilidad
Creación de procesos automatizados que apoyen al Sistema de Alta Disponibilidad
Actualización en línea del sistema
Configuración en línea de los servicios
Respaldos completos e incrementales
Recuperación ante fallo de *hardware* en uno de los nodos
Recuperación ante desastre

Tabla 6.8 Tópicos básicos para la capacitación al Grupo de Administradores

De esta capacitación surgirán nuevas definiciones de procedimientos y las que ya existían habrá que adaptarlas en términos de Alta Disponibilidad.

Una buena técnica que deberá implantarse es el uso de escenarios simulados, enfrentar al administrador ante una situación simulada puede ayudar a clarificar y llevar a la práctica todos los manuales de procedimientos que sean definidos. El hecho que un administrador sepa que hacer ante un problema y que tenga claramente los pasos que lo llevarán a su solución determinará el éxito del Sistema de Alta Disponibilidad.

Parte del resultado de la capacitación para los administradores, es la realización de la documentación detallada, es decir, la definición en forma escrita de todos los procedimientos que involucran tanto a los administradores como al Sistema de Alta Disponibilidad. Muchas veces este punto es olvidado, pero es de suma importancia para garantizar que las tareas del administrador estén bien realizadas.

La instalación y configuración del Sistema de Alta Disponibilidad, su administración, respaldos del sistema, configuración de servicios y el resto de los aspectos tocados en la Tabla 6.1 son susceptibles de ser definidos de manera escrita; por cada aspecto habrá que realizar al menos un manual de procedimientos.

CAPÍTULO 7

Conclusiones

CAPÍTULO 7: CONCLUSIONES

7.1 ¿EL OBJETIVO FUE ALCANZADO?

Recordando la Introducción del presente trabajo, el objetivo fundamental descrito fue el siguiente: **"IMPLEMENTAR UNA HERRAMIENTA DE SEGURIDAD CUYA MISIÓN SEA EVITAR PERDER EL SERVICIO ANTE CUALQUIER FALLA CAUSADA POR EL EQUIPO, SISTEMA OPERATIVO, APLICACIONES, FALTA DE ENERGÍA, DESASTRE NATURAL Y ANTE CUALQUIER FACTOR QUE ATENTE CONTRA LA DISPONIBILIDAD DEL SERVIDOR WEB".** He llegado al punto de evaluar dicho objetivo, si se cumplió o no, cuales son los factores críticos de éxito, entre otros.

Si analizamos el trabajo a partir del último capítulo hasta el primero, podemos darnos cuenta que implantamos la herramienta de seguridad (Capítulo 6) que fue desarrollada como parte de un Sistema de Alta Disponibilidad (Capítulo 5) el cual fue diseñado (Capítulo 4) a partir de los requerimientos precisos (Capítulo 3) que fueron obtenidos de un análisis de seguridad realizado teniendo como Objeto de Evaluación, el Servidor Web (Capítulo 2).

Por tanto puedo asegurar que mediante el desarrollo de este trabajo de tesis, se cumplió con el objetivo fundamental garantizando un nivel de disponibilidad cercano al 100% anual.

7.2 FACTORES CRÍTICOS DE ÉXITO

Como lo describo en el Capítulo 6, la implantación de la herramienta de seguridad, no sólo incluye la instalación del Cluster de Alta Disponibilidad y la Eliminación de Puntos Simples de Fallo, existen otros factores críticos de éxito, como son la participación activa de los recursos humanos, así como del apoyo de la Directiva, la existencia de manuales de procedimientos, capacitación constante, y sobre todo un cambio en la cultura administrativa de los servicios de Internet: Toda acción de los administradores u operadores, y en muchos casos de la Directiva, deberá estar pensada fundamentalmente en términos de Alta Disponibilidad.

En conclusión, puedo decir que el Sistema de Alta Disponibilidad, como sistema en sí, está compuesto por elementos que realizan una función en específico, trabajan para sí mismos y para el sistema en sí. Si falla un elemento, todo el sistema falla. El sistema desarrollado en este trabajo de tesis no sólo involucra el *hardware* y *software*, sino también al recurso humano y su cultura. Pienso que de poco sirve tener un sistema con nivel de disponibilidad tolerante a fallos, si uno de sus componentes, el factor humano, no está plenamente capacitado para enfrentar una situación en la cual el resto del sistema no puede actuar. Es muy importante el fomento de una cultura de Alta Disponibilidad, por ello considero que el Capítulo 6 deber tomado seriamente.

Otro factor crítico de éxito es el nivel de redundancia que realmente se implemente en torno al Objeto de Evaluación, durante el análisis y diseño del Sistema de Alta Disponibilidad, se dieron las pautas necesarias para alcanzar el 99.99% de Disponibilidad Anual del Servicio, para ello se diseño, por ejemplo, una configuración de red redundante, una configuración de discos duros redundantes, una configuración de servidores redundantes, etc. De acuerdo a los recursos y la decisiones de los administradores y Directiva se implementarán o no todos los niveles de redundancia sugeridos por el Analista-Diseñador de Sistemas de Alta Disponibilidad. Es decir, que los recursos y la disposición son otro factor crítico de éxito a tomar en cuenta.

Se sugiere que para garantizar aun más la Alta Disponibilidad de servicio, se coloquen los nodos del Cluster de Alta Disponibilidad lo más alejados posible, si se pudiera en distintos edificios o ubicaciones. Sin embargo una vez más caemos en la disponibilidad de recursos y disposición de la Directiva como factores críticos para el éxito del proyecto.

7.3 CONTRIBUCIONES

Considero que el desarrollo del presente trabajo produjo, en primera instancia a la Facultad de Ingeniería y a la comunidad informática mexicana en general, al menos una contribución importante: ***la iniciativa y fomento de la Cultura de la Alta Disponibilidad de los Servicios Informáticos***. Alrededor del mundo existe cantidad y calidad de información y soluciones relacionadas con la Alta Disponibilidad sobre todo en aquellos países que se consideran estar a la vanguardia tecnológica, México es un país considerado en vías de desarrollo, porque generalmente ligan la vanguardia tecnológica con las grandes inversiones y capital. Sin embargo, pruebo con hechos que, al menos en el campo de la Alta Disponibilidad en Servicios Informáticos, no se requieren de grandes montos de inversión para alcanzar niveles de disponibilidad muy cercanos al 100%, y mediante herramientas y Software Libre además de la creatividad se puede hacer mucho a tal grado de igualar o superar a las soluciones comerciales de los monstruos informáticos. Considero que la frase de hacer mucho con poco se aplica en este caso.

Contribuyendo con la iniciativa y fomento de este tipo de cultura, este trabajo condensa calidad en documentación relativa a la Alta Disponibilidad, ilustra un procedimiento o mecanismo para guiar a los Analistas-Diseñadores de Sistemas de Alta Disponibilidad, precisamente a eso, analizar, diseñar, desarrollar e implementar Sistemas de Alta Disponibilidad que garanticen sino el 100% de disponibilidad, sí un nivel cercano a tal cota máxima.

Indirectamente este trabajo de tesis contribuye a la creciente ola que inunda nuestro país: la necesidad de asegurar nuestros Sistemas Informáticos. No hace mucho tiempo, los administradores, operadores e inclusive la misma Directiva se preocupaban solamente porque funcionaran sus sistemas y ya, sin embargo

actualmente esto está incompleto, no solo es importante el aspecto funcional u operativo, sino el aspecto de seguridad también. La disponibilidad es un servicio de seguridad que debe ser fuertemente protegido contra cualquier amenaza que exista. Por ello es que considero el presente trabajo como una herramienta de seguridad para proteger dicho servicio de seguridad, fomentando así la conciencia que debemos de tener todos las personas que de alguna forma estamos involucrados en la Seguridad Informática de nuestros sistemas.

7.4 LÍNEAS FUTURAS

Como todo Sistema Informático es susceptible de ser mejorado día con día, éste no es la excepción. En particular, el Sistema de Alta Disponibilidad cumple con el objetivo planteado al inicio del trabajo, sin embargo puede ser mejorado a tal grado que sus resultados garanticen un nivel mucho muy cercano a la cota máxima de 100% de disponibilidad. Una línea futura sería esta, la de mejorar el desempeño de dicho sistema.

Por último quisiera agregar que considero que este trabajo podría impulsar a los Analistas-Diseñadores de Sistemas de Alta Disponibilidad a crear nuevos mecanismos que garanticen incluso niveles cercanos a la Tolerancia a Desastres que involucren pocos recursos, que optimicen recursos para ser más precisos para su elaboración.

APÉNDICE A

Fuentes de Información

FUENTES DE INFORMACIÓN

CAPÍTULO No.1

1.1 Weygant, Meter S., ***CLUSTERS for High Availability, A Primer of HP-UX Solutions***, Ed. Prentice Hall, pp. 166

1.2 SSAD: Sistema Simple de Alta Disponibilidad
Disponible en la Web:
< http://linux-labs.net/docs/backup-cluster.html >

1.3 Sistemas de alta disponibilidad bajo Linux
Disponible en la Web:
< http://www.linuxfocus.org/Castellano/November2000/article179.shtml >

1.4 Alta disponibilidad para Linux
Disponible en la Web:
< http://juampe.net/LinuxHA >

1.5 *Clustering* de Alta Disponibilidad bajo GNU/Linux
Disponible en la Web:
< www.bisente.com/documentos/clustering/informe.pdf >

1.6 Curso de doctorado de Alta Disponibilidad
Disponible en la Web:
< http://sweb.uv.es/~rmtnez/ha >

1.7 Configuración de Servicios de Internet de Alta Disponibilidad con Herramientas de Software Libre
Disponible en la Web:
< http://accusor.net/memorias_dsl/exposiciones/expo_simbiosis.html >

1.8 *High-Availability Linux Project*
Disponible en la Web:
< http://www.linux-ha.org >

1.9 *High Availability*
Disponible en la Web:
< http://www.linuxvirtualserver.org/HighAvailability.html >

1.10 *High Availability Linux with Software RAID*
Disponible en la Web:
< http://www.linuxjournal.com/article.php?sid=6412 >

1.11 Dossier de Alta Disponibilidad
Disponible en la Web:
< http://www.recursos-as400.com/dossier/ad >

1.12 Alta Disponibilidad. Marca la calidad del servicio.
Disponible en la Web:
< http://www.albasoft.com/docu/prd/ser_seg_disp.html >

1.13 Soluciones de Almacenamiento y Alta Disponibilidad
Disponible en la Web:
<http://www.getronics.com/es/eses/Knowledge/Documentacion/Folletos/Infraestructura/documentacion_nwp_almaltdisponibildad.htm >

1.14 *Service Level Agreement and SLA Guide*
Disponible en la Web:
< www.service-level-agreement.net >

1.15 *HISPACLUSTER*
Disponible en la Web:
<http://www.hispacluster.org/modules.php?op=modload&name=Web_Links&file=index&req=viewlink&cid=4 >

CAPÍTULO No. 2

2.1 *Common Criteria for Information Technology Security Evaluation*, January 2004, Version 2.2, Revision 256, CCIMB-2004-01-001
Disponible en la Web:
< http://csrc.nist.gov/cc/index.html >

2.2 Sitio Web de la Facultad de Ingeniería
Disponible en la Web:
< www.ingenieria.unam.mx >

2.3 Sitio Web de la Unidad de Servicios de Cómputo Académico
Disponible en la Web:
< www.ingenieria.unam.mx/~unica >

CAPÍTULO No.3

3.1 Sitio Web sobre las estadísticas del Servidor Web (Sitio Web restringido, solo personal autorizado)
Disponible en la Web:
< www.ingenieria.unam.mx/~estadisticas >

CAPÍTULO No. 5

5.1 Eric S. Raymond, ***The Art of UNIX Programming***, Ed. Addison-Wesley Professional, 3° ed., pp. 525.

5.2 Tanenbaum, Andrew, ***Redes de Computadoras***, Ed. Prentice may, 4° ed.

5.3 Kernighan, Brian W., Ritchie, Dennos M., ***EL LENGUAJE DE PROGRAMACIÓN C***, Ed. Pearson Educación, 2° Ed. pp. 294.

5.4 Sitio Web de *Fedore Core*
Disponible en la Web:
< fedora.redhat.com >

5.5 PORTSENTRY
Disponible en la Web:
< http://www.cert.unam.mx/Tutoriales/Tutoriales/portsentry/Portsentry.html >

5.6 SCANLOGD

Disponible en la Web:
< http://www.openwall.com/scanlogd >

5.7 LOGCHECK

Disponible en la Web:
< http://www.gwolf.org/seguridad/logcheck >

5.8 OPENSSH

Disponible en la Web:
< http://www.openssh.com >

5.9 POSTGRESQL

Disponible en la Web:
< http://www.postgresql.org >

5.10 APACHE

Disponible en la Web:
< http://www.apache.org >

5.11 PHP

Disponible en la Web:
< http://www.php.net >

5.12 OPENSSL

Disponible en la Web:
< http://www.openssl.org >

5.13 RSYNC

Disponible en la Web:
< http://samba.anu.edu.au/rsync >

5.14 Programación Básica de Sockets en Unix para Novatos

Disponible en la Web:
< http://es.tldp.org/Tutoriales/PROG-SOCKETS/prog-sockets.html >

CAPÍTULO No. 6

6.1 Plan de Contingencias y Seguridad de la Información

Disponible en la Web:
< http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5007/0300.HTM >

APÉNDICE B

Glosario

A

Acuerdo de Nivel de Servicio

Contrato entre el analista-diseñador de Sistemas de Alta Disponibilidad y los usuarios del sistema en el cual se determinan explícitamente los requerimientos de disponibilidad en términos de los servicios que son proporcionados a los clientes, así como también los recursos necesarios para proveer de manera satisfactoria dichos servicios.

Administrador

Es el personal encargado de la administración, operación, mantenimiento y seguridad de los Sistemas Informáticos.

Alta Disponibilidad

Caracteriza a un sistema que es diseñado para evitar la pérdida del servicio por reducción o administración de fallos así como la minimización de las interrupciones planeadas.

Amenaza

Todo aquello que intenta o pretende destruir.

Análisis de Riesgos

Término que hace referencia al proceso necesario para responder a tres cuestiones básicas sobre nuestra seguridad: ¿Qué queremos proteger? ¿Contra quién o qué lo queremos proteger? ¿Cómo lo queremos proteger?

Apache Web Server

Servidor Web HTTP/HTTPS de código fuente libre.

Arrays of RAID arrays

Arreglo de arreglos de discos RAID, es un conjunto que agrupa a dos o más arreglos de discos de tipo RAID.

Ataque

Es la realización de una amenaza a un Sistema Informático, se clasifican en ataques pasivos y en ataques activos. El primero consiste en el espionaje o robo de información, el otro tipo de ataque engloba toda acción que altera la información.

ATM

Asynchronous Transfer Mode, Modo de Transferencia Asíncrona es el corazón de los servicios digitales integrados que ofrecerán las nuevas redes digitales de servicios integrados de Banda Ancha.

Auditoria

Es la verificación de controles en el procesamiento de la información, desarrollo de sistemas e instalación con el objetivo de evaluar su efectividad y presentar recomendaciones a la Directiva.

B

Backup
Copia de seguridad o respaldo de información.

Banner
Medio por el cual se da a conocer cierta información como es el nombre del producto, versión, etc.

Beats
Latidos, forma en la cual se comunican los Nodos de un cluster.

Beowulf
Filosofía que enuncia las bases para la creación de Clusters de Alto Rendimiento, donde existe un Nodo Maestro y varios Nodos Esclavos interconectados a través de una red de área local..

Bridge
También llamado "puente" es aquel dispositivo de red más elemental y sólo permite conectar varias LAN de un mismo tipo.

C

Cableado Estructurado
Es cualquier sistema de cableado que permita identificar, reubicar y cambiar en todo momento, fácilmente y de forma racional los diversos equipos que se conectan, basándose en una normativa completa.

CGI
Common Gateway Interface, CGI es una especie de enlace entre el sistema operativo y el Servidor Web. Su función principal consiste en recibir una petición del servidor HTTP y ejecutar un programa usando los parámetros pasados por el servidor. Una vez que el CGI ha ejecutado el programa, regresa al Servidor Web la salida del programa.

Cluster
Conjunto de objetos. En el tema de servidores, se refiere a un grupo de servidores de trabajando de forma paralela en una misma tarea, los cuales aparecen como si fuesen un mismo servidor. Puede ser configurado para Balanceo de Carga o para Alta Disponibilidad.

Clustering
Se le llama así al conjunto de técnicas que permiten la creación de Clusters.

Common Criteria
Criterios Comunes. Estándar común de Criterios de Cómputo que garantizan la seguridad de los servicios, son versátiles, se pueden adaptar, norman los Servicios de Seguridad. En 1998 se publica la versión 2.0, en 2000 la versión 2.1 y en 2004 la versión 2.2. También es conocido como el estándar ISO-15408.

Concentrador
Dispositivo que permite centralizar el cableado de una red. También conocido con el nombre de hub. Un concentrador funciona repitiendo cada paquete de datos en cada uno de los puertos con los que cuenta de forma que todos los puntos tienen acceso a los datos. Son la base para las redes de topología tipo estrella.

Cron
Es el demonio calendarizador de tareas de UNIX/Linux, permite la ejecución automática de cierta tarea a un determinado tiempo.

Crossover Link
Cable UTP de tipo cruzado, que permite la interconexión de dos Nodos, en un extremo tiene configuración T568A y en el otro T568B.

D

Desfiguramiento de sitios Web
Tipo de ataque que consiste en la modificación o alteración del contenido de un sitio Web.

Detención
Conjunto de procedimientos que paran cierto servicio.

Dioscuri
También llamados Dioscuros o "Hijos de Zeus". Son dos hermanos, Castor y Pollux. La historia cuenta como Leda, su madre, en una misma noche, se unió a Zeus y a su esposo Tindáreo. De estas uniones puso dos huevos. De uno salieron Pollux y Helena, hijos de Zeus. Del otro huevo salieron Castor y Clitemestra, hijos de Tíndareo. Participan en numerosas leyendas, aun después de muertos.

Disk Druid
Es una herramienta de particionamiento de discos duros que viene incluido en la mayor parte de las distribuciones Linux.

Disk Mirroring
Técnica que permite redundancia en los dispositivos de almacenamiento. Consiste en replicar los datos de un dispositivo en otro garantizando que se preserven ante la falla de uno de ellos.

Downtime
Véase Interrupción.

Drive Duplexing
Técnica que involucra el uso del nivel RAID 1 complementado con el uso de dos controladoras de disco independientes.

Drive Spanning
Técnica que permite que varias unidades de almacenamiento sean vistas como una sola.

Drive striping
Técnica que se emplea para distribuir los datos entre las diferentes unidades de almacenamiento maximizando el rendimiento de las operaciones de entrada-salida. Para ello, se divide la unidad de almacenamiento en bloques de datos denominados bandas (stripes), las cuales se distribuyen entre los discos físicos. Durante las operaciones de lectura y escritura los discos operan simultáneamente.

Duplicidad

La duplicidad consiste en disponer en un mismo instante, dos instancias de un mismo objeto.

E

Equipo activo

Nombre genérico que reciben los dispositivos de red como son los *switches, hubs, bridges*, entre otros.

Esquema de Seguridad

Es el procedimiento que tiene por objetivo el asegurar los bienes de una institución mediante la determinación de los Requerimientos de Seguridad que garanticen dicho objetivo, esto se hace a través de un análisis del entorno de Seguridad de los bienes a asegurar, determinando las amenazas a las que están expuestos, además de las Políticas y los Objetivos que las anulen.

Ethernet

Norma o estándar (IEEE 802.3) que determina la forma en que los puestos de la red envían y reciben datos sobre un medio físico compartido que se comporta como un bus lógico, independientemente de su configuración física.

F

Failover

Término genérico que se usa cuando un nodo debe asumir la responsabilidad de otro nodo, importar sus recursos y levantar el grupo de servicios.

Failure

Fallo. Se define como la suspensión de la operación normal de algún componente, los fallos ocurren a nivel hardware, software, administración de sistema y red, inclusive en el ambiente.

Fault Tolerant

Sistemas Tolerantes a Fallos, *Fault tolerant* o FT. En los cuales, el servicio está siempre activo. Todos los recursos necesarios están replicados para ofrecer una disponibilidad continua en caso de fallo.

FiberChannel

Es un estándar de la industria adoptado por la ANSI y la organización ISO, específico para la conexión y transporte de datos. Soporta tasas de transferencia de hasta 2 gigabit por segundo, tasa cinco veces más rápido que SCSI. *Fiber Channel* soporta protocolos diversos y transmisión de datos para los más populares estándares de red y canal tales como SCSI, HIPPI, *Ethernet*, IP y ATM, combinando las características de unos con otros.

Filtrado de Paquetes

Conjunto de funciones que permiten o deniegan el acceso de Aplicaciones de Internet hacia a la red privada, basándose en los puertos TCP/UDP.

Firewall

Un *Firewall*, cortafuegos o pared de fuego, es un equipo de hardware o implementación en software utilizado en las redes de computadoras para prevenir algunos tipos de comunicaciones prohibidas por la política de red.

Firmware

Es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM. Al estar integrado en la electrónica del dispositivo es en parte hardware, pero también es software, ya que proporciona lógica y se dispone en algún tipo de lenguaje de programación.

Frame Relay

Frame Relay ha sido diseñado principalmente para hacer más eficiente el transporte de datos entre localidades remotas que requieren intercambiar grandes cantidades de información a altas velocidades de transmisión utilizando de manera óptima el ancho de banda.

G

Gateway

Es un dispositivo, frecuentemente una computadora, que realiza la conversión de protocolos entre diferentes tipos de redes o aplicaciones. Por ejemplo, un *gateway* de correo electrónico, o de mensajes, convierte mensajes entre dos diferentes protocolos de mensajes.

Gestor de Arranque

Programa empleado para elegir en el arranque o encendido de un sistema, el sistema operativo con el cual se quiera trabajar en el caso de tener más de uno instalado.

Giveaway

Véase Switchover.

GNU

GNU es un acrónimo recursivo que significa "GNU No es Unix". El sistema GNU fue diseñado para ser totalmente compatible con UNIX. Para asegurar que el software GNU permaneciera libre para que todos los usuarios pudieran "ejecutarlo, copiarlo, modificarlo y distribuirlo", el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos al tiempo que evitase restricciones posteriores de los mismos.

GPL

General Public License o Licencia Pública General. Licencia creada por la *Free Software Foundation* y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.

GRUB

GRUB es probablemente el gestor de arranque más potente para máquinas basadas en procesadores i386. Tiene un menú bastante cómodo como interfaz e incluso soporta la operación a través de una consola serie.

Grupo de Recursos

Son todos los recursos necesarios para que el servicio sea ofrecido a un cliente.

Grupo de Servicios
En Alta Disponibilidad, se denomina Grupo de Servicios a un determinado servicio ofrecido a un cliente.

H

HA
Véase Alta Disponibilidad.

Hearbeat
Es el vínculo de comunicación entre nodos que el software de Alta Disponibilidad establece en un Cluster de Alta Disponibilidad.

High Availability
Véase Alta Disponibilidad.

Hipótesis
Conjunto de suposiciones de como debiera de operar seguramente el sistema en estudio.

Home Directory
Es el directorio de trabajo de los usuarios de los sistemas UNIX/Linux.

Host
Término que reciben los equipos o máquinas generalmente.

Hot Plug Operations
Son todas aquellas actualizaciones o reparaciones de hardware mientras el sistema esta ejecutándose.

Hot-spare
Técnica aplicada cuando el hardware de RAID detecta que ha perdido uno de los discos útiles, lo reemplaza inmediatamente por el de reserva e inicia la reconstrucción, de forma que el tiempo que permanecemos sin tolerancia a fallos es mínimo y no se requiere en ningún momento una intervención manual para restablecer las condiciones de fiabilidad iniciales.

Hot-swap
Técnica con la cual podemos reemplazar el disco en línea sin necesidad de apagar el sistema. Una vez que hemos introducido el nuevo disco inmediatamente comienza la reconstrucción del mismo.

HTML
HTML, *Hyper Text Markup Language* o Lenguaje de Marcado de Hipertexto. Es un lenguaje informático diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

HTTP
HTTP es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan *Hyper Text Transfer Protocol*, es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas Web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página Web, y la respuesta de esa Web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares.

HTTPS
Versión segura del protocolo HTTP. El sistema HTTPS utiliza un cifrado basado en *Secure Socket Layers* (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP.

Hub
Véase Concentrador.

I

IDS
Intrusion Detection System o Sistema Detector de Intrusos. Es un conjunto de aplicaciones que están a la escucha de alguna intrusión en el sistema.

Interrupción
Se define como la duración de un paro del servicio.

Intrusión
Sucede cuando una entidad no autorizada entra ilegalmente a un sistema y es capaz de utilizarlo y modificarlo como si fuera un usuario legítimo.

Inyección de código arbitrario
Técnica que consiste en ejecutar instrucciones inválidas a los programas tratando de romper su estabilidad.

Iptables
Es una serie de reglas que se emplean para aceptar o denegar un paquete mediante un módulo del *Kernel* de Linux.

J

Journaling
Con *journaling* se lleva una cuenta de que se ha ido modificando en el sistema de archivos, ya que a la hora de verificarlo sólo comprobara las inconsistencias de unos pocos archivos y directorios

K

Kernel
Es el núcleo o *core* del sistema operativo.

L

LAN
Local Area Network. Red de Área Local.

Linux

Linux estrictamente se refiere al núcleo Linux, pero es más comúnmente utilizado para describir un sistema operativo similar a Unix pero libre, también conocido como GNU/Linux, formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU. La primera versión del núcleo Linux fue escrito por el hacker finlandés Linus Torvalds y liberado en 1991, combinado con componentes de GNU.

Linux Fedore Core

Es una distribución del núcleo del Sistema Operativo Linux personalizada por el *staff* de Fedora.

Log

Es un registro de un suceso o actividad.

Logcheck

Software de Seguridad que se encarga de analizar las bitácoras generadas por el sistema, eliminando toda información "no crítica", y elaborando un reporte de lo que si considera crítico.

Logical Host

Unión de Grupo de Servicios y Grupo de Recursos asociados a éste.

Login

Sinónimo de Nombre de usuario.

Logwatch

Software de Seguridad que funciona de forma parecida a Logcheck.

M

MD5

Software que permite realizar *Checksums* o sumas de comprobación. Se puede emplear para garantizar la integridad de un archivo.

Mirror Server

Servidor replica del servidor original. Mediante diversas técnicas se garantiza un porcentaje cercano al 100% de replicación o *mirrorring.*

Mirroring

Véase replicación.

MOSIX

Es un paquete de software que mejora el *kernel* de Linux con capacidades de computación de Clusters. El *kernel* mejorado permite cualquier Cluster de estaciones de trabajo y trabajar en coordinadamente como parte de un sólo sistema.

MTTF

El Tiempo Promedio Entre Fallos o *Mean Time Between Failure* mide el tiempo promedio transcurrido hasta que un dispositivo falla.

MTTR

El Tiempo Promedio de Recuperación o *Mean Time To Recove* que mide el tiempo promedio tomado en restablecer la situación normal una vez que se ha producido el fallo.

N

NFS

Significa *Network File System* o Sistema de Archivos en Red, permite compartir y explotar un Sistema de Archivos en la red.

NIS

Significa *Network Information Service* o Sistema de Información en Red, también es conocido como YP o *Yelow Pages* es una herramienta para simplificar la administración de varias máquinas en red. Te permite compartir y centralizar *passwords*, direcciones de *hosts,* etc

Nodo

Un nodo es un sistema el cual es miembro de un Cluster.

O

Objetivo de Seguridad

Son todos aquellos elementos que contrarrestan las amenazas a un Sistema Informático.

Objeto de Evaluación

Es aquel ente central alrededor del cual se desarrolla el Esquema de Seguridad.

On-line Backups

Respaldos en Línea, es una técnica que permite realizar Copias de Seguridad sin necesidad de manifestar una interrupción planeada del servicio, de esta forma mientras dure esta actividad en sistema sigue en línea.

OpenSSH

Permite establecer conexiones remotas de manera segura puesto que la información viaja de manera cifrada entre cliente-servidor

OpenSSL

Librerías y herramientas para asegurar la confidencialidad de la información que viaja por Internet.

Outage

Paro. Se define como la pérdida específica de un servicio que es percibido por el usuario. No todos los fallos causan paros y no todos los paros no planeados son causados por fallos.

P

Parche

Actualización del software.

Particionamiento

Acción de dividir en porciones definidas una unidad de almacenamiento.

Periodo Normal de Operación
Se define como aquel lapso en el cual se espera que el sistema este operando y ofreciendo un servicio de calidad al usuario.

PHP
Hypertext Preprocessor, es un lenguaje de programación de *scripts*, concebido en el tercer trimestre de 1994 por Rasmus Lerdorf. Se utiliza principalmente para la programación de CGIs para páginas Web, destaca por su capacidad de ser embebido en el código HTML.

Plan de Contingencias
Es la estrategia a seguir ante un desastre natural que atente contra la disponibilidad e integridad del sistema en producción.

Política de Seguridad
Lineamiento cuyo cumplimiento garantiza la seguridad del bien o servicio en cuestión.

Portsentry
Detecta una conexión a un puerto no válido o un barrido de puertos y toma acciones de bloqueo contra el *host* hostil

PostgreSQL
Servidor de base de datos de código fuente libre.

Proxy
El término *proxy* hace referencia a un programa que realiza una acción en representación de otro.

Punto Simple de Fallo
Es un eslabón débil que forma parte de una cadena de elementos que conforman un Sistema Informático.

PVM
Parallel Virtual Machine o Máquina Virtual Paralela es un software que permite ver al grupo de computadoras que integran el Cluster como si fuera una sola máquina de procesamiento.

Q

Query o queries
Generalmente reciben este término todas las consultas y actualizaciones a una base de datos.

Quorum
Elemento de decisión que le permite a los Nodos de un Cluster saber que hacer en dado caso de que no tengan comunicación. El Nodo que tenga en su poder el *quorum* es el que se queda con el Grupo de Servicios.

R

RAID
Una técnica para proveer redundancia es el uso de arreglos de discos con configuraciones RAID o *Redundant Array of Inexpensive Disks*. Existen varios niveles de RAID y cada uno tiene sus particularidades.

RDSI

Red Digital de Servicios Integrados RDSI o ISDN se define como una red que procede por evolución de la Red Digital Integrada (RDI) y que facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados.

Recovery

Conjunto de técnicas que posibilitan la recuperación del sistema ante una falla.

Redundancia

Término fundamental en la Alta Disponibilidad, que consiste en proveer más de una alternativa, más de un camino o más de un elemento asegurando su disponibilidad.

Requerimiento de Seguridad

Conjunto de elementos que definirán lo que hay que hacer para garantizar la seguridad de un sistema.

Requerimientos Específicos de Rendimiento

Es el conjunto de requerimientos del cliente en cuanto a como le gustaría que funcionara el sistema durante el Periodo Normal de Operación.

Root

Comúnmente se le llama "Superusuario", y es aquel entidad que tiene plena jurisdicción dentro del Sistema Operativo. A esta cuenta accede el Administrador para desempeñar sus funciones. La seguridad de esta entidad es crítica.

Router

Un *router*, enrutador o encaminador es un dispositivo de interconexión de redes de computadoras que opera en la capa 3 (nivel de red) del modelo OSI. Este dispositivo interconecta segmentos de red o redes enteras. Hacen pasar paquetes de datos entre redes tomando como base la información de la capa de red.

Rsync

Esta herramienta permite no sólo realizar una copia de la información, sino que permite tener una sincronización de ella, de esta forma todos los cambios en la información de un nodo se replican de una forma muy sencilla. Si esta información viajara en formato plano a través del medio, sería inseguro y atentaría contra la confidencialidad de la misma, por ello, RSYNC implementa un mecanismo de cifrado durante su acción.

Runlevel

Nivel de ejecución del Sistema Operativo Linux.

S

SAN

Storage Area Network o Red de almacenamiento local, es una red local de alta velocidad con el fin de tener dispositivos de gran volumen para tener un almacenamiento masivo, donde cada servidor o máquina conectada a esta red tiene acceso a los medios de almacenamiento.

Scanlog

Es un monitor de barrido de puertos, es decir, detecta e informa vía *log* de un barrido de puertos de un *host* hostil

Script

Conjunto de instrucciones que ejecutan como los programas de procesamiento por lotes de MSDOS.

Scripting, Lenguaje de

Lenguaje donde la ejecución de un programa no pasa por el proceso de compilación, sino que al momento de ejecutarse lo hace línea a línea y si encuentra algún tipo de error, para la ejecución.

SCSI

SCSI es un protocolo para intercomunicación entre la CPU, la tarjeta madre y los dispositivos periféricos de entrada-salida. SCSI significa *Small Computer System Interface* o Sistema de Interfaz para Computadora Pequeña. También en SCSI se esta preparando un sistema en serie, que además es compatible con SATA, esto es, se podrán conectar discos SATA en una controladora SAS (serial Attached SCSI).

Service Level Agreement

Véase Acuerdo de Nivel de Servicio.

SFP

Security Function Policy, Política de Seguridad funcional.

ShellScript

Lenguaje de programación de tipo *Scripting*. Un script o programa ShellScript condensa una serie de comandos que el SHELL interpreta y ejecuta.

Simmetrix Remote Data Facility

Symmetrix Remote Data Facility o SRDF duplica la información del sitio en producción en uno o más sistemas *Symmetrix* separados físicamente, independientemente de su ubicación.

Single Points Of Failure

Véase Punto Simple de Fallo.

SOIT

Sum Of Inoperative Times o Suma de los tiempos de inoperatividad.

Splitbrain

Un *Splitbrain* o división de cerebros, es un caso especial de *failover*, en el cual falla el mecanismo de comunicación y gestión de un cluster de dos nodos. Es una situación en la cual cada nodo cree que es el único activo, y como no puede saber el estado de su nodo compañero, tomará acciones en consecuencia, forzando un *takeover*.

SPOF

Véase Single Points Of Failure.

SPU

System Processor Unit. El SPU contiene uno o más procesadores, memoria, y una fuente de poder. EL SPU es un componente de un nodo, por tanto, un nodo no es lo mismo que un SPU.

SRDF

Véase *Simmetrix Remote Data Facility*.

SSH

Secure Shell. Véase *OpenSSH.*

SSL/TLS

Secure Sockets Layer /Transport Layer Security. Véase *OpenSSL.*

Stripe

Son bloques o bandas de datos

Swap

Partición de intercambio que emplea el sistema como memoria física virtual.

Switch

Un *switch* es un dispositivo de interconexión de redes de computadoras que opera en la capa 2 (nivel de enlace de datos) del modelo OSI. Este interconecta dos o más segmentos de red, funcionando de manera similar a los puentes.

Switchover

Es un *failover* manual, consiste en ceder los recursos de un grupo de servicios a otro nodo del cluster, mientras se realizan ciertas tareas administrativas.

T

Takeover

Es un *failover* automático se produce cuando un nodo nota un fallo en el grupo de servicios. Para ello debe haber cierta monitorización con respecto al grupo de servicios. El Nodo que se declara averiado es forzado a ceder el servicio y recursos.

TCP/IP

El nombre TCP/IP proviene de dos protocolos importantes de la familia, el *Transmission Control Protocol* (TCP) y el *Internet Protocol* (IP). Todos juntos llegan a ser más de 100 protocolos diferentes definidos en este conjunto. El TCP/ IP es la base del Internet que sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local y área extensa. TCP/IP fue desarrollado y demostrado por primera vez en 1972 por el departamento de defensa de los Estados Unidos, ejecutándolo en el ARPANET, una red de área extensa del departamento de defensa.

TCP_WRAPPERS

Es un monitor de paquetes, que trabaja con los archivos /etc/hots.allow y /etc/hosts.deny. Máquinas que se encuentren el /etc/hosts.deny no pueden realizar ciertas conexiones y las incluídas en /etc/hosts.allow si pueden hacerlo

TET

Total Elapsed Time o Total del Tiempo Transcurrido.

TNOF

Total No. Of Failures o Número total de fallos o interrupciones.

TOE

Target Of Evaluation. Véase Objeto de Evaluación.

TOT

Total Operating Time o tiempo total de operación.

TSF

TOE Security Functions. Funciones de Seguridad de la TOE.

U

Uninterruptible Power Suply

Fuente de Energía ininterrumpible, mejor conocido como *No-Break*. Ofrece un respaldo de energía ante un servicio intermitente de la energía eléctrica.

UNIX

Es un sistema operativo portátil, multitareas y multiusuarios; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

UPS

Véase *Uninterruptible Power Suply*.

Uptime

Tiempo de operatividad.

V

Vulnerabilidad

Todos aquellos huecos de seguridad que tienen los sistemas. Fallos en el sistema.

W

Web Server

Servidor de contenido estático o dinámico que puede ser visualizado e interactuar con el mediante un navegador modo texto o modo gráfico.

Webalizer

Es un analizador de bitácoras Web, que genera informes y estadísticas sobre la actividad del servidor.

WWW

La World Wide Web, la Web o WWW, es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador Web para extraer elementos de información de los Servidores Web y mostrarlos en la pantalla del usuario.