Capitulo

1 INTRODUCCIÓN

La era informática se ha ido desarrollando a pasos agigantados, llegando a ofrecer soluciones de alto nivel de desempeño, sin embargo, muchas veces existe aun la dependencia de elementos humanos que deben intervenir para activar un sistema o iniciar un proceso de recuperación en caso de falla La alta disponibilidad de Sistemas define que en caso de que un sistema computacional falle, sus datos deben ser recuperados en una cantidad de tiempo razonable. La definición de "Razonable" varía ampliamente dependiendo del tipo de industria que esta operando Sin embargo es de alta relevancia que el concepto sea aplicado en toda empresa que le dé importancia a su información

Hubo un tiempo en que los conceptos de alta disponibilidad se asociaban con los bancos, y equipos mainframe fault-tolerant de precios excesivos para una empresa mediana o pequeña. Hoy las organizaciones pueden recibir el mismo nivel de disponibilidad a un costo mucho menor haciendo uso de sistemas abiertos.

La alta disponibilidad se basa en una mezcla de soluciones concertadas para remediar fallas en discos, aplicaciones, redes, sistemas operativos, y computadoras que afecten el flujo de información y operaciones que componen la base de una compañía.

Conforme se hacen diversos análisis, podremos ver que hay soluciones que pueden tomarse como obvias, pero que mientras no se haga este listado de posibles causas de fallas, no han sido visibles para nadie Por ejemplo, puede ser obvio que mi red de área local puede fallar en

cualquier momento por lo que debería tener instalaciones alternas de red con tarjetas alternas y duplicar nodos de conexión, sin embargo, esto no se contempló en un principio por el costo que podria haber representado.

Una empresa especifica puede tener diseñados planes de contingencia que le permitan continuar con las operaciones teniendo los sistemas fuera de servicio (por fallas, mantenimientos, o respaldos), con un costo en la productividad Sin embargo, estos planes de contingencia que siempre deben existir deben ser lo últimos en aplicarse, siempre debe existir una solución de alta disponibilidad de sistemas, aunque esto puede implicar adquisición de más equipos y personal más preparado.

Para que una solución de Alta Disponibilidad tenga éxito en un largo plazo, se requiere del compromiso de las Gerencias y del personal de Sistemas de Información para ir evolucionando en su ambiente operacional, enfocando los esfuerzos en las personas, procesos y la tecnología. Además la empresa debe establecer métricas que indiquen el cumplimiento del requerimiento de Alta Disponibilidad: ¿Será suficiente que el sistema garantice su operación al 96%? ¡Esto sería aproximadamente 2 semanas al año sin servicio por fallas!

Es importante tener bien claro las necesidades propias de la empresa o sector para determinar como proporcionarle una solución real y factible Si tus operaciones dentro de la empresa son de enfoque científico, muy difícilmente podrás hallar una solución de alta disponibilidad

1.1 Hipótesis

"No existe una metodología establecida para el desarrollo de proyectos de alta disponibilidad, por lo que basado en mi experiencia, y con el fin de apoyar futuros proyectos de Alta Disponibilidad propondré como parte de mis conclusiones una metodología para el desarrollo de los proyectos de alta disponibilidad"

1.2 Objetivos del Proyecto

Se pretende hacer un análisis de los distintos conceptos de alta disponibilidad que existen actualmente, investigando requerimientos tecnológicos, humanos y económicos requeridos para su aplicación.

Basándose en estos conceptos, se presenta un ejemplo de evaluación costo / beneficio de la implementación de un proyecto de alta disponibilidad.

Clarificar que la alta disponibilidad no significa que el sistema este operando el 100% del tiempo

Que impacto tiene para una empresa de comercio electrónico el que su sistema de comercialización en línea este fuera de servicio.

El crecimiento de los costos de implementación y administración de un sistema en alta disponibilidad según se incrementa el porcentaje de disponibilidad ofrecido

El tiempo máximo al año que un sistema en alta disponibilidad puede estar fuera de servicio según el porcentaje de disponibilidad esperado Algunos niveles de disponibilidad no podrán ser garantizados aún con la tecnología actual

Mostrar que partes de un sistema impactan más en la disponibilidad del mismo.

La importancia que la alta administración este involucrada y consciente del costo que implica para la empresa el no tener un sistema en alta disponibilidad

Los sistemas se deben organizar de acuerdo al nivel de disponibilidad esperado, no todos los sistemas se deben configurar con el mismo nivel de disponibilidad, deberán clasificarse por nivel de impacto dentro del negocio contra el costo de la inversión a realizar.

Explicar que un esquema de recuperación de desastres es un nivel de Disponibilidad Continua y Alta Disponibilidad más avanzado y que implica requerimientos más complejos los cuales no son el enfoque de esta tesis

1.3 Alcance

Esta es una la evaluación de los esquemas de alta disponibilidad en el manejo de cluster On-Site: Los esquemas de recuperación de desastres y operación de Sites Remotos pertenecen a un rubro de estudio por si mismos, por lo que fueron excluidos de este material.

Se presenta un procedimiento general que podría seguirse para la implementación de un esquema de alta disponibilidad La información técnica solo se detalla lo necesario para enfocar el tema y clarificar los pasos usuales que deben seguirse para esta implementación Este procedimiento general es solo una posible guía de apoyo para iniciarse en el tema.

Debido a la alta dinámica de los mercados de aplicaciones y software, los productos aquí mencionados pudieran haber quedado desactualizados con respecto a la fecha en curso Por la misma causa, los productos mencionados no son todos los que existen, pero si nos ayudan a ubicar las áreas comunes entre ellos así como las diferencias básicas entre los mismos

Como existe suficiente material sobre cada producto como para definir un proyecto por si solo, el material presentado prescinde de profundidades altamente técnicas, principalmente para poder hacer una revisión rápida de las semejanzas y diferencias entre los productos, tal que nos ayude a formar una idea más clara de los conceptos de alta disponibilidad.

1.4 Antecedentes

¿Quién puede necesitar de una Alta Disponibilidad en sus Sistemas? La respuesta es todo aquel que se precie de darle importancia a su negocio. Todo negocio que trabaje en un esquema de 7x24 o que no pueda sufrir de cortes de servicio por un tiempo mayor de 2 minutos o hasta un máximo de 30 minutos (según la definición de alta disponibilidad que tenga la empresa)

En 1995 dos investigaciones llevadas a cabo por Oracle Corporation y por Datamation mostraron que los negocios habían perdido en promedio entre 80,000 y 350,000 dólares por hora debido a fallas no planeadas En 1993 la bomba que se colocó en el World Trade Center provocó que 145 de los 350 negocios que se ubicaban en este edificio cerraran por no contar con una infraestructura redundante de Sistemas que les permitiera continuar las operaciones o recuperar la información de sus transacciones.

Con base en estos datos, es claro que el configurar y mantener una infraestructura redundante de Sistemas es un bajo precio comparado contra la pérdida del negocio por no haber hecho este esfuerzo Además, habrá de considerarse que los tiempos fuera (downtime) de los sistemas pueden deberse a fallas no planeadas así como a mantenimientos preventivos. Si contamos con esquemas de Alta Disponibilidad, estos mantenimientos se podrán realizar sin afectar la continuidad de los servicios en ningún momento

Conforme las empresas de todos los tipos continúan implementando sistemas de mayor complejidad y poderío, la disponibilidad de las aplicaciones, hacia los usuarios a través de la organización, se ha convertido en algo importante

La ecuación de disponibilidad involucra un número de elementos que deben funcionar como parte de un todo: La red, la plataforma de hardware, sistema operativo, y software de aplicaciones. Históricamente, los responsables de los departamentos de Sistemas se han enfocados en la plataforma de hardware, sin evaluar que, sin un desempeño confiable del sistema, la disponibilidad de la aplicación sería imposible. Al mismo tiempo, conforme las infraestructuras de redes se convierten en parte vital de las organizaciones, se incrementa la atención prestada al desempeño de ruteadores, hubs y elementos específicos de la red.

Investigaciones realizadas por IDC (International Data Corporation) muestran que las organizaciones de clientes y usuarios están volviéndose más conscientes de que un sistema con mayor tiempo de operación y una red más confiable no garantizan por sí solo la disponibilidad máxima de la aplicación.

Existen varios detonantes que presionan para lograr mejores esquemas de alta disponibilidad

La Internet – Quizás uno de los mayores detonantes de los requerimientos de alta disponibilidad, ha empujado a muchas organizaciones de todos los tamaños y de todas las industrias hacia una dependencia de los servicios de información y tecnología en un esquema 7x24x365 (Las 24 horas de la semana y los 365 días del año.) Hoy con la revolución del e-comerce que está cambiando ***literalmente*** los modelos de negocio hacia negocios que operen por la noche, los clientes buscan proveedores que operen bajo la WWW (World Wide Web) Muchas empresas que tienen cierto número de clientes en el día (en América) probablemente lograrán otro número nuevo de clientes por la noche (En Europa y Asia) . Esto no solo se aplica a empresas comerciales, sino también empresas de manufactura que pueden recibir pedidos via Internet de productos de "materia prima" para procesamiento adicional por parte de un cliente, ciertamente en estos casos primero habrá una negociación previa para que el cliente y el

proveedor puedan abrir una línea de crédito, pero una vez establecida, será una muy natural forma de levantar pedido, aún cuando el personal laboral de la empresa no esté en oficinas en ese momento, muchos sistemas de pedidos, basándose en los datos preconocidos del cliente enrutan y programan pedidos para su procesamiento en forma automática.

Integración de los procesos de negocio a la tecnología de información. Al igual que las necesidades de disponibilidad para Internet, los procesos de negocio se integran más y más a la tecnología de información, hasta niveles en que no se distingue entre el proceso y la aplicación que lo habilita. Por ejemplo, la habilidad de la organización de almacenar y recuperar la información de los clientes Anteriormente se manejaban los legajos y gabinetes, ahora es mediante herramientas de datawarehouse y minería de datos que se ha logrado una transformación total de está actividad Adicionalmente con esta transformación, la importancia de esta actividad se ha incrementado, hasta niveles en los cuales, una implementación pobre, puede llevar a una compañía a una perdida competitiva real

Globalización de los mercados y negocios. Otro de los factores que están demandando de mejores niveles de disponibilidad, es la rápida globalización de los negocios y las organizaciones En la actualidad, muchos negocios tienen sistemas de información distribuidos, un sistema de servicio de intranet cuyo servidor este ubicado físicamente en Hong Kong al fallar puede afectar las operaciones de la compañía en lugares como París, Nueva York, o Los Ángeles Tal nivel de impacto era impensable hace algunos años, cuando aún las compañías multinacionales realizaban instalaciones de sistemas en forma regional, local o nacional, concentrando posteriormente la información mediante procesos de consolidación.

Las nuevas tecnologías requieren de la mas alta disponibilidad de los sistemas y aplicaciones de la compañía para garantizar la competitividad. Uno de los factores más impactantes es el mercado electrónico el ***e*-business**, es precisamente en esta nueva área tecnológica donde él

dejar de ofrecer servicio, a causa de un sistema fuera (downtime) que, implica grandes pérdidas. No solamente se pierde un cliente que llega al sitio de Internet a buscar un producto que comprar, sino que además este cliente seguramente se irá a un competidor, si este le dio un buen servicio, seguramente este cliente ya nunca regresará. Pero además, si este cliente comenta como adquirió ese producto y como **nuestro** sitio no le dio el servicio, y como el competidor inmediatamente lo atendió, seguramente hemos perdido al menos otras 10 ventas. Multipliquemos esto por el número de clientes que no pudieron acceder a nuestro sitio en este mismo momento.

Algunas investigaciones han mostrado que las empresas están invirtiendo en combinaciones de sistemas de alta disponibilidad con sistemas identificados de misión crítica Se espera que para finales del 2002 la demanda de servicios de alta disponibilidad se incremente. En Estados Unidos de Norteamérica, se ha presentado el siguiente comportamiento.

Rubro	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**
Hardware						
Inversion en Soporte de Hardware de mision critica	646	719	774	857	926	995
Inversion total en hardware	12,663	12,399	12,288	12,240	12,187	12,133
Porcentaje de la inversion en Mision Critica	5.1%	5.8%	6.3%	7.0%	7.6%	8.2%
Software						
Gastos de soporte en software de mision critica	553	702	885	1,101	1,348	1,626
Gasto total en software	8,782	10,172	11,806	13,591	15,674	17,869
Porcentaje de la inversión en mision critica	6.3%	6.9%	7.5%	8.1%	8.6%	9.1%
Total						
Inversion total en soporte a la mision critica	1,199	1,421	1,660	1,958	2,274	2,621

Total invertido	21,445	22,571	24,094	25,831	27,861	30,002
Porcentaje de la inversión en misión crítica	5.6%	6.3%	6.9%	7.6%	8.2%	8.7%

Tabla 1. Inversiones para el soporte de aplicaciones de misión crítica:[1]
Nota. Estos gastos no incluyen las inversiones hechas en infraestructura y dispositivos de alta disponibilidad en servicios específicos de redes (millones de dólares.)

Los vendedores de hardware han realizado inversiones cuantiosas para proporcionar servicios de alta disponibilidad, pero con tecnología propietaria Con la llegada de nuevos sistemas operativos como ocurre con Windows NT, los proveedores de hardware (HP, IBM, Compaq y otros) se han visto forzados a desarrollar soluciones de alta disponibilidad que no nacen en sus plataformas y que no son desarrollados por ellos mismos Ahora esta ocurriendo que estos proveedores de hardware son empujados a competir entre ellos, con un sistema operativo que es elaborado por un tercero (Microsoft)

Existen productos de terceros que en forma independiente del sistema operativo Windows NT 2000, pero basándose en asociaciones estratégicas han desarrollado métodos de ofrecer soluciones de alta disponibilidad, uno de estos ejemplos, es la solución ofrecido por Oracle con su producto "Real Application Clusters", el cual pretende ofrecer una solución de disponibilidad continua con recursos propios, no dependientes directamente del S O

[1] **International Data Corporation** high availability Not just for hardware anymore an IDC whitepaper

Capitulo

2 MARCO TEÓRICO

2.1 Conceptos de alta disponibilidad y requerimientos para su funcionalidad

Se hará una referencia a las definiciones y conceptos básicos que nos permitirán profundizar en el tema de la alta disponibilidad.

2.1.1 ¿Que es la Alta Disponibilidad?

Minimización del número y duración de ocurrencias planeadas o no planeadas de suspensión de servicio de los sistemas que soportan la operación de una compañía, por mantenimientos preventivos, o por fallas de los mismos

Nivel de Servicio es el periodo esperado de servicio disponible y tiempo aceptable de servicio no disponible.

2.1.2 Definición de Alta Disponibilidad

Un sistema que es diseñado, puesto en práctica y desplegado con componentes suficientes para satisfacer las exigencias funcionales del sistema, pero el que también tiene la redundancia suficiente en componentes (el hardware, el software y procedimientos) para enmascarar ciertas fallas definidas, tiene Alta Disponibilidad (HA) Esta definición es ambigua. los

términos(condiciones) "suficientes", "enmascarar" "ciertas" requieren una clarificación más profunda En vez de hacer esto, sin embargo, debemos enfatizar que debido a esta ambigüedad, existe un gran número y clases de configuraciones que con esta definición pueden ser clasificadas como de "Alta Disponibilidad".

Definamos ahora con más detalle los términos(condiciones) ambiguos

- **El Enmascaramiento** de una falta implica el proteger de la observación externa de la falla. Este acercamiento es el equivalente computacional del adagio filosófico " Si un árbol se cae sin nadie para oírlo, este no hace ningún sonido ". El enmascaramiento es una técnica "de juego de manos" para asegurar no el hecho de que la falta ocurra, sino que esta no sea observable

  Recordemos que las faltas son definidas como una desviación inesperada del comportamiento especificado El enmascaramiento de una falla significa que ninguna desviación (o más precisamente que las desviaciones definidas) del comportamiento especificado ocurre Esto invariablemente se logra mediante un mecanismo de réplica apropiado al componente, una estrategia de redundancia. Cuando un componente falla, el componente redundante lo sustituye El grado de transparencia en la que este reemplazo ocurre puede conducir a una amplia variación de los sistemas que se llaman de "Alta Disponibilidad" Tenemos el siguiente espectro de enmascaramiento existentes:

  - **Manual Masking: Enmascarado Manual(MM)**. Después de una falla de un componente, se requiere alguna acción manual para poner el componente redundante en servicio, durante este tiempo el sistema no estará disponible para el su uso La expectativa usual es que la HA implica una recuperación automatizada

De ahí, los sistemas que usan "el enmascaramiento manual" generalmente no son considerados HA)

- **Cold Standby (CS) réplica parcial (o sustituto en frió)** Después que un componente falla, los usuarios del componente son desconectados y pierden cualquier trabajo en progreso (esto es, ellos regresan la transacción operada a un pasado consistente, y estable de su trabajo) Un mecanismo automático de detección de falla y recuperación descubre la falla, y ponen en servicio el componente redundante Este componente redundante ha permanecido inactivo y deberá ser inicializado para entrar en servicio. Una vez que esto es hecho, los usuarios son capaces de continuar su proceso desde el punto hasta donde se regresó su transacción Típicamente el tiempo requerido para que el proceso de detección de fallas descubra esta falla e invoque el componente redundante es bastante bajo (decenas de segundos) Sin embargo, el tiempo requerido para la inicialización del componente redundante puede ser mucho más largo. Este tiempo de recuperación es dependiente de la aplicación, pero por lo general implica la limpieza de los filesystems, bases de datos y otros recursos persistentes hacia un estado consistente de información, que fácilmente puede tomar decenas de minutos.

- **Warm Standby (WS) réplica en Caliente (o sustituto parcial).** Después de que un componente falla, los usuarios del componente son desconectados y pueden perder parte de su trabajo en progreso. El mecanismo automático de detección y recuperacion de fallas descubre la falla y notifica al componente redundante para que asuma la operación. Este componente redundante ha estado corriendo activamente y esta parcialmente inicializado Además, este puede ya estar compartiendo algo del estado de procesamiento de su par fallado. De ahí, no

necesariamente deberá reiniciarse todo el trabajo en progreso. Los clientes del componente todavía deben unirse activamente al nuevo componente redundante Los tiempos de detección de falla para los sistemas en Warm Standby son similares a los que tienen los sistemas en Cold Standby, pero los Tiempos de recuperación son dramáticamente más cortos que en el CS (típicamente algunas decenas de segundos), debido a la inicialización parcial y estado operacional compartido.

- **Hot Standby (HS)/Active Replication(AR) Réplica en Caliente/Replicación Activa (o Sustituto en Caliente/Replicación Activa)**. Los Componentes activos y Redundantes están fuertemente acoplados en grupos y son (lógicamente) indistinguibles a los usuarios del grupo de componentes En realidad, el usuario no "ve" el grupo sino sólo el comportamiento requerido del componente. El Estado de Procesamiento es compartido activamente y completamente entre los componentes de grupo. Después de una falla de uno de los componentes del grupo, los usuarios del componente no son desconectados y no observan la falla de ningún modo El trabajo en progreso sigue con el(los) componente(s) en redundancia que restan en el grupo que proporciona la funcionalidad del componente En este modelo, el enmascaramiento es completo y transparente - los clientes del sistema no son interrumpidos

  Los tiempos de recuperación son instantáneos – con más exactitud, el concepto de tiempos de recuperación no se aplicaría, puesto que desde la perspectiva del cliente no hay ninguna recuperación El término "Réplica Activa" se prefiere sobre "Reserva en Caliente", puesto que éste último hace referencia a una relación asimétrica entre un componente "Activo" y uno "De reserva" (o en espera) concepto utilizado en los términos "Reserva en Frío" y "Reserva Parcial" El término

"Réplica Activa" refleja mejor, sin embargo, la simetría entre las replicaciones. Vamos a referirnos a esto como Reserva en Caliente/Replicación Activa (HS/AR).

- **La Suficiencia** es una reflexión de las exigencias del sistema para la Alta Disponibilidad (¡una definición recurrente!) Por ejemplo, un sistema diseñado para apoyar la tolerancia de fallas de hardware sólo podría enmascarar fallas de hardware, pero no fallas de software. Esto sería "suficiente" para las exigencias de aquel sistema, y tal sistema no enmascararía fallas de aplicación

  La práctica Aceptada, sin embargo, es usar el término " Tolerancia a Fallas " (Fault-Tolerance) para tales sistemas de solo-enmascarado-de-hardware, mientras el término " Alta Disponibilidad " es reservado para el sistema que enmascara fallas en el hardware, en el software y en los procesos Siguiendo esta convención, " la redundancia suficiente " en la definición de Alta Disponibilidad anterior implica que deben enmascararse tanto las fallas del hardware, del software y de los procesos

  Además de la determinacion de que las fallas son enmascaradas, La Suficiencia también refleja cuantas veces son enmascaradas Por ejemplo, un acercamiento de par-empatado reproduce cada componente exactamente una vez, esto es "suficiente" para soportar(resistir) un solo punto de falla Por otra parte, un sistema en el que cada componente tiene n>2 replicaciones puede sobrevivir más fallas simultáneas (donde "simultáneo" implica que las fallas ocurren dentro de la ventana de reparación del primer componente que falló)

- **La Certeza**, como se usa en la definición de HA, se reconoce del hecho que no todas las faltas pueden ser enmascaradas, por esta razón se debe establecer con certeza que fallas **si son** enmascaradas, cualquier otra falla no establecida con certeza, no sería enmascarada

definitivamente Por ejemplo, algunos errores de diseño, y que son reproducibles, raras veces son enmascarados. Así un sistema que se supone, autentifica a un usuario antes de permitir el acceso, pero cuyo diseño no lo estableció correctamente, bajo ningún nivel de replicación se corregirá esta falta, y con algunos intentos repetidos los usuarios hostiles podrán fácilmente exponer la falta y lograr el acceso.

Existen técnicas como la programación n-way para poder atacar esta clase de faltas, sin embargo, estas técnicas raras veces se usan debido a su complejidad y costo. De ahí que se reduce el juego de fallas que serán compensadas en un sistema Alta Disponibilidad. Este conjunto de restricciones, puede conducir a una amplia variación de sistemas que se llaman a sí mismos de "Alta Disponibilidad".

2.1.3 Acrónimos y Abreviaciones[2]

Existen términos, acrónimos y abreviaciones que se usan con frecuencia para referirnos a temas de alta disponibilidad, la tabla 2 menciona algunos de los mismos.

Acrónimo	Nombre	Definición
BA	Basic Availability Disponibilidad Básica	Un sistema cuya ingeniería ofrece un servicio funcional, pero que no hace ninguna prevencion para atrapar las fallas.
CA	Continuous Availability (Disponibilidad Continua)	Enmascara completamente los cortes de servicio planeados o no planeados La Alta Disponibilidad HA es un subconjunto de la disponibilidad continua (CA) La disponibilidad continua asume el uso estricto del modelo de replicación activa Hot Standby (Copia de Sustitución caliente)
HA	High Availability	Enmascarado automático de cortes de servicio no planeados, puede

[2] Acronyms And Abreviations from "A Modern Taxonomy of High Availability"

	(Alta Disponibilidad)	implementarse mediante replicación en frío, incremental o en caliente.
MM	Manual Masking (Enmascarado Manual)	Concepto de seudo disponibilidad (muy débil) que utiliza la replicación mediante rutinas dentro del sistema, pero que requiere detección manual y acciones de recuperación Debido a esto no cumple con las condiciones de HA (Alta Disponibilidad) de reactivación automática.
CS	Cold Standby (Sustituto en Frío)	Es la más débil forma de Alta Disponibilidad, se generan réplicas completas de la instalación, pero no se tiene la información actualizada, por lo que al presentarse una falla el servicio reinicia con una fuerte perdida de información.
WS	Warm Standby (Sustituto Parcial)	Forma moderada de alta disponibilidad, se generan réplicas de la información y se actualizan parcialmente, y pueden tener algunos estatus de procesamiento en preparación para una falla.
HS	Hot Standby (Sustituto en Caliente)	Esta es el modelo más fuerte de Alta Disponibilidad, las réplicas son copias en línea y que comparten el estatus con la instalación primaria. De hecho, no se puede diferenciar directamente entre la instalación primaria y la instalación de recuperación. Esto hace que el término "primario" sea más a forma de referencia que otra cosa.
AR	Active Replication (Replicación Activa)	Técnica usada para obtener la Disponibilidad proporcionada por la Replicación en Caliente (o Sustituto en Caliente) "Hot Standby".
LR	Passive (Lazy) Replication (Replicación Pasiva)	Técnicas aplicadas para obtener un Sustituto Parcial del sistema (Warm Standby).
MFP	Make Forward	Termino aplicado a sistemas con Replicación Activa (Hot Standby)

	Progress (Avanzar la Operación)	que acentúa que los clientes y usuarios de los sistemas en operación no pueden verse interrumpidos en sus operaciones o actividades en el evento de una falla, y que estos deben continuar operando normalmente sin perder un solo cambio de su operación.
RR	Rollback and Recover (Regreso y Recuperación)	Término aplicado a los sistemas con replicación parcial o en frío Warm/Cold Standby que enfatiza que los usuarios y clientes de un sistema que dejo de operar sufrirán una interrupción mínima de servicio, y deberán regresar algunas de sus operaciones hasta un estado de procesamiento que sea consistente antes de reiniciar el proceso de recuperación de la falla.
MTTF	Mean Time To Failure (tiempo medio para fallar)	Promedio de vida de un componente que operará hasta fallar.
MTTR	Mean Time To Restore (Tiempo medio de recuperación)	Tiempo promedio requerido para reparar un componente y reactivar el sistema, o tiempo promedio requerido para restaurar el servicio despues de una falla
MTBF	Mean Time Between Failures (Tiempo promedio entre fallas)	Es el tiempo transcurrido desde que se inicia el servicio hasta que se reinicia nuevamente el servicio MTBF MTTF+MTTR
	Availability (Disponibilidad)	A MTTF / (MTTR + MTTF)
HVAC	Heating, Venting and Air Conditioning	Equipo de calefacción, ventilación y enfriado

	equipment	
UPS	Uninterruptable Power Supply	Fuente ininterrumpible de poder

Tabla 2. Algunos acrónimos y abreviaciones de alta disponibilidad

2.1.4 Algunos términos

Cluster: Existen diversas definiciones de cluster, algunas son:

1) Conjunto de servidores configurados para proporcionar los servicios computacionales, y recibir la carga operacional en caso de que uno de los nodos falle.
2) Conjunto de servidores configurados para ofrecer servicios computacionales en grupo donde cada nodo ofrece un servicio especifico y funcional, y el grupo ofrece un servicio total
3) Conjunto de servidores organizados para balancear la carga de trabajo entre ellos, y poder ofrecer servicios computacionales con un alto nivel de desempeño.
4) Arreglo de Servidores dedicados a atender peticiones específicas de operación, y servidores redundantes que se encuentran en estado de espera para soportar la carga de trabajo en caso que el servidor primario llegará a fallar

Los Tipos de cluster existentes serían:

- **Cluster Elástico y Escalable**: Servidores acoplados en forma sencilla, y que contiene múltiples sistemas unidos mediante un esquema de balanceo de cargas. Son cluster escalables horizontalmente. Los sistemas no interfieren ni consideran a otros sistemas en el mismo cluster. Las instancias de aplicación se administran en forma autónoma, con transacciones y juegos independientes de datos, que se replican a los nodos en forma individual a través de la

red, o vía un servidor NFS. Los nodos se administran en forma individual, apoyado mediante prácticas y procedimientos locales que facilitan su manejo.

- **Cluster para Performance** Este tipo de clusters es típico para sistemas computacionales de alto desempeño (HPC), que se enfocan en el desempeño y escalabilidad de sistemas al aplicar tantos procesadores (CPU) como sean posibles para resolver un problema o cálculo específico La mayoría de los clusters científicos usan alguna forma de procesamiento por lotes, o software de trabajo compartido Este tipo de clusters no tiene capacidad elástica, en caso de una falla de la aplicación o sistema, debe existir un sistema de verificación del nivel de avance y su mecanismo de recuperación, para poder reiniciar el procesamiento de los lotes que han fallado

- **Cluster para Alta Disponibilidad** Este tipo de clusters agrega capacidades de alta disponibilidad, al montarse sobre la infraestructura del sistema operativo. La disponibilidad se logra al usar scripts que monitorean la sanidad de los nodos individuales del cluster En caso de falla en los servicios (debido a fallas en discos, redes, o los servicios de la aplicación misma) los recursos y las aplicaciones serán asignados y reiniciados en otro nodo. Los nodos individuales del cluster, se administran principalmente en forma independiente uno del otro.

Failover Evento en el cual el cluster reubica una aplicación de un nodo que ha fallado hacia un nodo sano, perdiéndose las operaciones que se estaban llevando a cabo en el nodo fallado, el otro nodo del cluster adquiere el control de los recursos y levanta los servicios del nodo fallado para soportar los requerimientos operacionales Los clientes del cluster pueden ver una ligera interrupción en los servicios, pero no se deberán darse cuenta del cambio de servidor

Failback Evento donde el servidor primario de un cluster reinicia sus operaciones retomando nuevamente el control de los recursos y levanta los servicios que un nodo secundario estaba proporcionando a causa de un failover. Esto es, regresar los servicios al servidor que esta identificado como proveedor primario de los mismos.

Escalabilidad. Habilita que un servicio cumpla niveles crecientes de carga, al mismo tiempo que se entrega la misma calidad de servicio Una aplicación escalable hace uso de los múltiples nodos en un cluster al correr varias instancias de los mismos servicios de aplicación Permite así mismo que a un nodo se le incremente su capacidad (en una ventana de mantenimiento), mientras los otros nodos cubren el servicio que este nodo debe proporcionar, al terminarse el mantenimiento este nodo toma sus recursos y ofrece mejores niveles de servicio Al nivel aplicativo, la escalabilidad se proporciona mediante el paralelismo, donde las transacciones individuales pueden ejecutarse en paralelo en varios nodos del cluster a la vez, esto requiere que el software de la aplicación soporte el paralelismo y sea responsable de sincronizar los datos. Cada instancia de la aplicación sería responsable de atender

Elasticidad· Capacidad de la aplicación, o sistema de resistir las fallas, soportando la migración de actividades hacia otro nodo de un cluster, con el menor tiempo de corte de servicio, y con el menor efecto visible hacia el usuario de un sistema.

2.1.5 Como medir la Disponibilidad esperada[3]

En un nivel simple, la disponibilidad, ya sea alta, baja o media se puede medir como una parte del tiempo que un servicio se tiene operando normalmente Es decir, el período de tiempo que el

[3] Chapter II What is resiliency from Blueprints of High Availability 1st Edition, Marcus Evans & Halt Stern Editorial John Wiley & Sons, Inc

sistema está realmente disponible durante el tiempo que debería estar disponible, se puede expresar mediante la fórmula:

Disponibilidad = MTTF / (MTTF+ MTTR)

Donde MTTF es el tiempo transcurrido para fallar (mean time to failure) y MTTR es el tiempo promedio de resolución de la falla (mean time to restore).

Aquí podemos observar que.

1. Según el MTTR se acerque a cero, la disponibilidad se acercará al 100%.
2. Conforme el MTTF (Tiempo transcurrido para fallar) se incremente, el MTTR tendrá un impacto menor en la medida de Disponibilidad

Por ejemplo, si un sistema particular tiene un MTTF de 100,000 hora, y el MTTR es de 1 hora, esto nos da un nivel impresionante de Disponibilidad de 100,000/100,001, o un total de 99 999 por ciento. Si se reduce el MTTR a 6 minutos (o 1 décimo de una hora) la disponibilidad se incrementa a un 99.9999 por ciento. Pero para lograr este nivel de disponibilidad requerirías de componentes cuya duración real entre fallas fuera de 100,000 horas que es alrededor de 11.4 años Dicho de otra forma: para lograr un porcentaje de 99.9999 de disponibilidad, cualquier parte o componente específico deberá durar al menos 11.4 años sin fallar y cuando falle solo dispondrás de un máximo de 31.5 segundos por año (o 6 minutos en 11 4 años) para recuperarte de la falla. Pero estos 6 minutos sin servicio son para todo el sistema, no sólo para un componente específico Con la tecnología actual, esto es inalcanzable y no realista Quizás sea más realista pedir que los tiempos máximos de resolución de fallas sean de 10 minutos al año, lo cual nos da un total de 99 998 % de disponibilidad y lo cual sea probablemente más alcanzable. Pero será muy difícil subir de este valor

A continuación se muestra la relación[4] entre el nivel de disponibilidad (cuantos nueves) y el impacto de tiempo máximo fuera de servicio hasta recuperarse de la falla

Porcentaje de Disponibilidad	Porcentaje de Tiempo fuera	Tiempo Fuera x Año	Tiempo Fuera x Semana
98%	2%	7.3 Días	3 horas 22 minutos
99%	1%	3.65 Días	1 hora 41 minutos
99.8%	0.2%	17 horas 30 minutos	20 mins. 10 Segundos
99.9%	0.1%	8 horas 45 minutos	10 mins. 5 segundos
99.99%	0.01%	52 mins 30 segundos	1 minuto
99.999%	0.001%	5.25 minutos	6 segundos
99.9999%	0.0001%	31.5 segundos	0.6 segundos

Tabla 3. medida del nivel de disponibilidad

2.1.6 Términos Relevantes a la discusión de la Alta Disponibilidad

Para el tema que nos concierne es necesario definir algunos términos y así poder comprender el enfoque de la alta disponibilidad, y allanar el camino hacia el desarrollo de esta tesis

[4] Table 2 1/Chapter II Measuring Availability from Blueprints of High Availability 1st Edition, Marcus Evans & Halt Stern Editorial John Wiley & Sons, Inc

2.1.6.1 Sistema en Alta Disponibilidad[5]

Cuando nos referimos a un sistema en HA, hacemos mención no solo del programa de aplicaciones y operaciones que se utiliza, sino, a una amplia gama de componentes que lo conforman y que soportan el todo como una sola unidad, algunos de estos son tomados en cuenta para el diseño del modelo de HA y otros definitivamente no son involucrados en los mismos, ya sea por su complejidad, por su irrelevancia, o simplemente porque el factor costo / beneficio no es suficiente para justificarlo.

Los componentes usualmente incluidos como parte de un sistema en HA, muchas veces no fueron necesariamente hechos bajo el enfoque de HA (Alta Disponibilidad), pero que deberemos considerar en el diseño de un sistema en HA

1) Hardware del Servidor que se requiere bajo una implementación distribuida
2) Discos y dispositivos de almacenamiento asociados al servidor
3) Software de aplicaciones implementado como parte del ambiente configurado de servidor.
4) Software de Sistema Operativo y para servicios de comunicación, y dominios
5) Conexiones de Red (Lan y Wan) que permiten un ambiente distribuido,
6) El hardware de escritorio (Clientes PC o Workstations) usado en un ambiente de trabajo distribuido
7) El software de aplicaciones implementado en el hardware de escritorio
8) El software de sistema operativo y herramientas de soporte que se implementan como parte del ambiente del equipo cliente (Herramientas GUI, utilerías de acceso a la informacion, etc).

[5] Definitions from "A Modern Taxonomy of High Availability"

9) Comunidades de usuarios (personas) que utiliza las herramientas GUI para operar las aplicaciones existentes en el equipo servidor.
10) La comunidad administrativa(personas) que administran y mantienen el sistema implementado y a su comunidad de usuarios
11) Los procedimientos administrativos, las políticas y lineamientos de seguridad utilizados en un ambiente operacional (Respaldos, Almacenamiento histórico, administración de perfiles de usuario, etc.)

Los componentes que usualmente no son incluidos en el diseño de un sistema en HA son:

1) El ambiente para la construcción del modelo de HA, que se compone de los ambientes de definición y creación usados para configurar el comportamiento del sistema, pero que no formarán parte final del mismo cuando quede ya en producción, estos ambientes se consideran temporales y no se plantean como parte de la solución final
2) Sistemas heredados (o sistemas preexistentes) con los cuales el Sistema habrá de interactuar pero que no son parte de él per se Estos sistemas pueden tener sus propios diseños de HA pero no están coordinados con el diseño del Sistema actual
3) Elementos y comunidades de personas que no interactúan con el sistema Por ejemplo, clientes del negocio que interactúan con los usuarios de los sistemas pero que no tocan directamente al sistema directamente, así como las áreas de administración y planeación.

Ver la figura 1 para evaluar que es lo que si se incluye usualmente en los modelo de sistemas de Alta Disponibilidad, y que es lo que no se incluye usualmente en estos modelos

2.1.6.2 DownTime (El Corte de Servicio)

La definición de DownTime varía mucho desde una definición amable, hasta una definición estricta, y desde sencilla hasta compleja Una definición estricta puede implicar que se tiene un DownTime cuando la red esta lenta o el servidor tiene bajo desempeño, o simplemente cuando un sistema no esta operando

La mejor definición la dan Marcus Evans y Hal Stern en su libro "Blueprints of High Availability" y es la siguiente: ***Si un usuario no puede lograr hacer su(s) tarea(s) a tiempo a causa del sistema, entonces, el sistema esta abajo, o ha fallado (es decir, tenemos un Downtime).***

Figura 1. Las distintas causas de fallas de sistemas

Existen múltiples causas de Cortes de Servicio, como se muestra, sin embargo, de los cortes de servicio el 30% corresponde a mantenimientos y actualizaciones de aplicaciones, sistema operativo, software crítico, o quizás es readecuación de arreglos, reorganización de los datos, etc otras ocasiones corresponde a depuración de logs, directorios y archivos temporales y limpieza de memoria

Muchos de los mantenimientos de hardware con la tecnología de muchos proveedores, se pueden hacer en línea, sin dar de baja los servicios Arreglos con discos reemplazables en

caliente, algunas aplicaciones se pueden remplazar en línea. Algunas veces se mueve la operación a otro equipo en un esquema de cluster para la alta disponibilidad

Otro factor es el recurso humano, a veces el usuario no sabe como debe realizar alguna tarea específica y comete algún error drástico que provoca la caída de la aplicación, la mejor forma de combatir estos errores es incluir en el diseño del sistema en HA un desarrollo amigable y sencillo de usar, así como un plan de educación y entrenamiento continuo sobre el manejo del sistema, así como el establecer como un rubro de este diseño, la necesidad de documentación solidamente preparada y siempre a la mano.

La parte más sorprendente en el DownTime es aquel relacionado con las fallas de Hardware. Este DownTime representa sólo el 10% del total de los cortes de servicios. Es decir, que con el mejor arreglo de discos (RAID) y un gran número de discos, tarjetas de red, y CPU's redundantes, así como el mejor hardware de redes del mundo, solamente estaríamos previniendo la ocurrencia del 10% de los cortes de servicio totales[6].

El factor de mayores causas de corte de servicio es el software de aplicaciones que esta operando, los bugs en el software de aplicaciones son los más difíciles de eliminar del sistema.

[6] Causes of Downtime, Chapter II What is resiliency, Blueprints of High Availability, Evans Marcus & Hal Stern

2.1.6.3 Faltas, Fallas y su relación con DownTime

Una falta es una desviación del comportamiento esperado de un sistema. Es decir, si el sistema

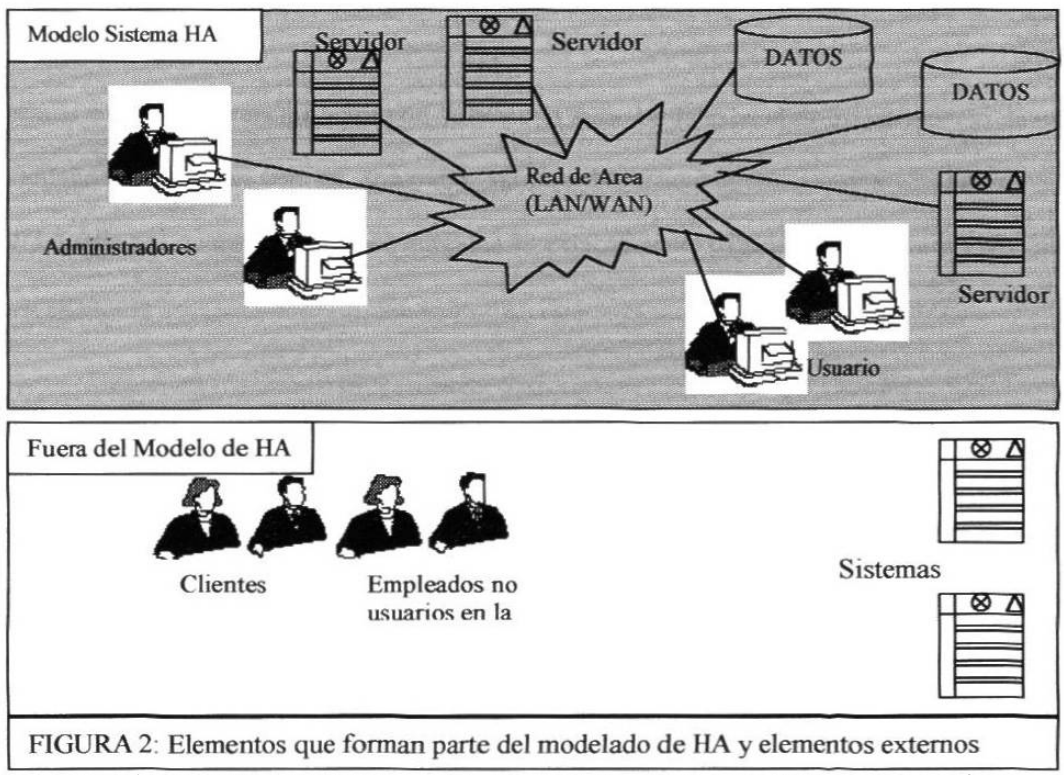


FIGURA 2: Elementos que forman parte del modelado de HA y elementos externos

está configurado para proporcionar y exhibir una funcionalidad muy específica, y en el proceso de ejecución el sistema produce una diferencia funcional que es palpable, entonces una falta ha ocurrido. El sistema proporciona la funcionalidad requerida mediante un procedimiento contenido en el software que se ejecuta en un hardware que involucra Equipos cliente, servidores, redes, almacén de datos, y otros periféricos Las faltas pueden presentarse en los procedimientos, software o hardware y pueden clasificarse como reproducibles o no reproducibles

Faltas Reproducibles: Un conjunto prescrito de pasos que lleva a observar la presentación de la falta en forma predecible.

Faltas no reproducibles La aparición de la falta se presenta en forma aleatoria o no determinada, o esta asociada a un origen o raíz que se encuentra fuera del sistema que se ha implementado.

Debemos siempre diferenciar entre una falta y una falla. La falta es el no-cumplimiento de una funcionalidad dentro del sistema, que puede ser palpable o no palpable externamente al usuario final La falla por su parte es aquella falta que es externamente palpable como una interrupción del servicio.

Ejemplos de faltas

- Error de diseño de la aplicación, la funcionalidad no es como se espera.
- El usuario debe ser validado pero esto no ocurre.
- El cálculo de un valor debe ser `a+b` pero se codificó para efectuar `a-b`
- Una falta es un acceso erróneo a un valor en memoria (siempre y cuando no provoque la caída del sistema)

Ejemplos de fallas·

- La ocurrencia de una falta que efectúa una acceso erróneo a un valor en memoria y que provoca la caída del sistema.
- Una falla en un dispositivo de hardware Disco duro, tarjeta de red.
- Una falta de diseño que provoca que se sobrecargue de procesos un servidor, provocando una falla del sistema y caída del mismo

Las fallas se clasifican en diversos ramos, algunas se denominan fallas fuertes· Son fallas que al correr un conjunto de pasos se presentan en forma idéntica y en la misma forma. Otras se denominan fallas Suaves. Son fallas que al correr un conjunto de pasos pueden presentarse ocasionalmente y otras no. La alta disponibilidad es muy útil y permite atacar a las fallas suaves, pero no es tan eficiente con las fallas duras.

2.1.6.4 Disponibilidad Básica

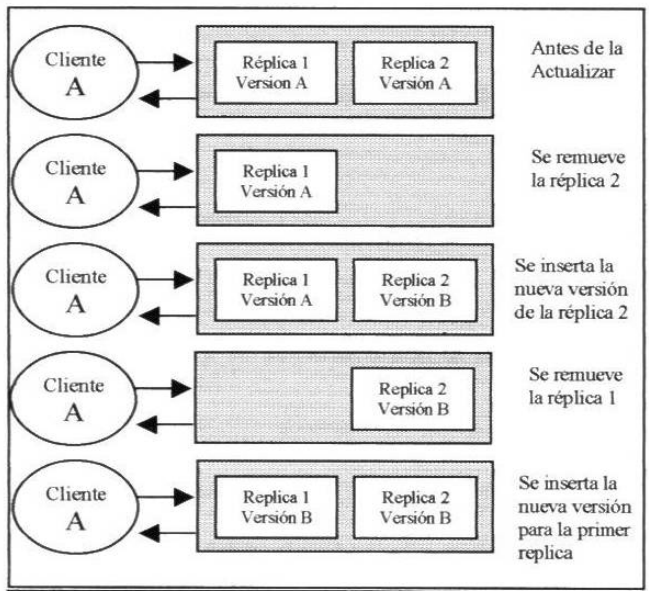


Figura 3: Pasos para lograr la disponibilidad básica

Un sistema que es diseñado, puesto en práctica y desplegado con componentes suficientes (el hardware, el software y procedimientos) para satisfacer las exigencias funcionales del sistema, pero no más, tiene la Disponibilidad Basica (BA) Tal sistema entregará la funcionalidad correcta mientras que no ocurran faltas / fallas y no se realicen operaciones de mantenimiento, etc. Siempre que una falta ocurre o una operación de mantenimiento es realizada, sin embargo, se observa una interrupción del servicio. Los sistemas de Disponibilidad Típicamente Básicos son desplegados como simples sistemas (no reproducidos)

2.1.6.5 Disponibilidad Continua

La Disponibilidad Continua (CA) amplia la definición de Alta Disponibilidad, sobre fallas no planeadas, y lo aplica a interrupciones planeadas también. Consistiendo entonces en un sistema que enmascara ambas (interrupciones imprevistas así como interrupciones planeadas) Los

Sistemas Continuamente Disponibles deben tener una estrategia de enmascarar que se ocupa de interrupciones planeadas Implicando una definición más vigorosa, mientras que la HA permite una variedad de estrategias para enmascarar (reserva en frío, reserva parcial y reserva en caliente), el sistema CA se limita a operar exclusivamente con el modelo de "Reserva en Caliente / Réplica Activa" - el enmascaramiento transparente debe ser completo, no solamente para fallas, sino para interrupciones planeadas también.

Para reconocer las dificultades inherentes a esto, consideremos las implicaciones para una mejora de software:

- Al principio, dos procesos idénticos son las réplicas uno del otro.

- Para realizar una mejora, una réplica se saca de servicio, y se substituye con la nueva versión mejorada.

- Al ponerse en línea la nueva versión en la réplica 2 debe absorber el estado de procesamiento de la versión 1.

- Posteriormente el segundo par (la réplica 1) se da de baja y la réplica 2 con la nueva versión toma el control de los clientes.

- Finalmente, el sistema totalmente replicado se recrea con ambas copias actualizadas a la versión B. Por supuesto que para alcanzar esto, las versiones A y B deben ser lo suficientemente compatibles para compartir el estado de procesamiento entre ellos. También, los clientes (que típicamente tienen un ciclo operativo independiente,

representado como la Versión X), deben ser capaces de inter funcionar transparentemente a través de ambas versiones A y B de servidor.

Todas estas cuestiones de compatibilidad dependen del estado de procesamiento de la aplicación y el comportamiento esperado, no hay soluciones "fáciles" para estas exigencias. De hecho, si la actualización de la Versión A hacia la B es bastante significativa (por ejemplo una gran cantidad de nuevas funcionalidades adicionales se implementan en la Versión B), esto no será posible en lo absoluto

2.1.6.6 Dominios para la Alta Disponibilidad

La definición de los componentes que un sistema en alta disponibilidad debe contemplar, no implica que todos estos componentes se repliquen para poder obtener de ellos una disponibilidad alta o continua En su lugar, el sistema deberá decomponerse para identificar y establecer cuales componentes tendrán disponibilidad básica, alta o continua Adicional a esto, si el componente del sistema estará bajo el esquema de alta disponibilidad es necesario identificar si la replicación será con enmascaramiento manual, reserva en frío, reserva parcial, o reserva en caliente / replicación activa. De esta forma podremos seccionar los sistemas en HA en dominios de disponibilidad, por las siguientes razones

- Porque algunos tipos de componentes por su costo prohibitivo no son tan factibles para establecerse en disponibilidad continua, o aun en alta disponibilidad

- Porque algunos tipos de componentes por implicaciones (o inexistencia) de la tecnología no son tan factibles para la disponibilidad requerida

- Porque el desempeño global del sistema en HA a se vería tremendamente degradado si un componente específico se pone en disponibilidad alta o continua.

- Porque el diseño para soportar la disponibilidad requerida es demasiado complejo y la implementación puede acarrear riesgos adicionales de la operación

- Porque sólo se justifica disponibilidad continua para algunos componentes (como discos o arreglos de discos) y varias formas de alta disponibilidad (o aún disponibilidad básica) para otros componentes.

Las tareas para el diseño de la alta disponibilidad se orientarían entonces a

1 Identificar la tecnología a utilizar para cada una de las variaciones de disponibilidad MM, CS, WS, HS/AR, la disponibilidad básica o BA se asume que no requiere una tecnología adicional a la configuración misma del sistema.

2 Basándose en el criterio anterior, establecer cuales componentes del sistema en HA deberán estar en BA, MM, CS, WS, HS/AR (o CA)

3 Habremos de organizar los componentes en los dominios y ubicar las tecnologías que permiten soportar este dominio.

En la figura siguiente se trata de ejemplificar esto, vemos que para en el dominio de disponibilidad continua, solo puede haber componentes que estén todos bajo el rubro de disponibilidad continua. Sin embargo, para el dominio de alta disponibilidad, algunos componentes estarán en reserva en caliente, otros en reserva parcial, y algunos otros componentes proporcionarían la alta disponibilidad mediante el esquema de reserva en frío. En el dominio de disponibilidad básica, tenemos algunos componentes con enmascarado manual y otros que definitivamente no tienen forma de tener un par de reemplazo (o reserva).

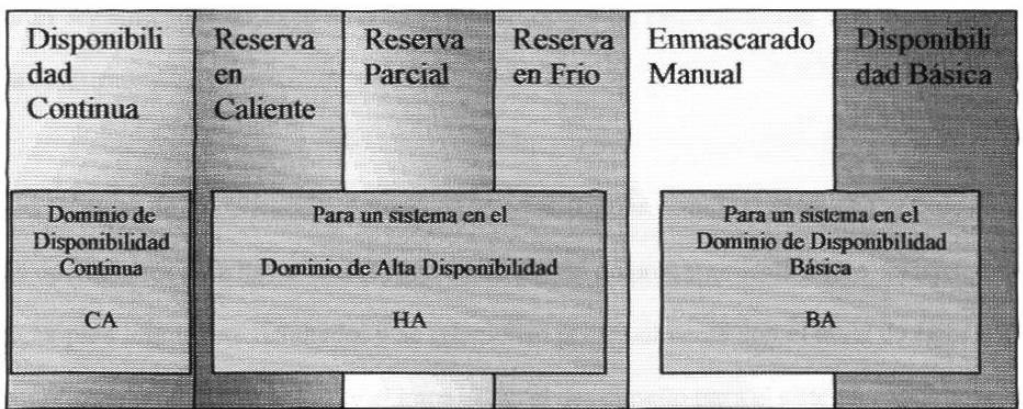


Figura 4: Los distintos niveles de disponibilidad y como se clasifican

2.1.6.7 La Replicación Activa

Anteriormente la replicación activa se manejaba como un sinónimo de la Reserva en Caliente, sin embargo, la Replicación Activa se usa también para describir el conjunto de técnicas que se utilizan para lograr implementar esta Reserva en Caliente, al compartir activamente el estado de procesamiento entre ambas réplicas Como el estado de procesamiento es un elemento dinámico en un ambiente distribuido, el compartir el estado de procesamiento entre las réplicas implica que no solo se transfiere información entre ambas, también deberán coordinar y sincronizar esta información Esto para que las réplicas puedan presentar hacia el exterior un estado estable, e internamente se procese en forma consistente

En general, los cambios de estado de procesamiento son no conmutativos, es decir, el aplicar dos cambios de estado de {A,B} a un momento de cambio S tal que: $S \rightarrow S_A \rightarrow S_{AB}$ lleva a un estado de procesamiento final distinto al obtenido si se aplican los dos cambios {B,A}: $S \rightarrow S_B \rightarrow S_{BA} <> S_{AB}$.

De aquí que el sincronizar un estado de cambio, en una réplica, significa que las operaciones deberán ser aplicadas en exactamente el mismo orden de procesamiento que se aplicaron a la copia maestra.

Así las técnicas de "Replicación Activa" están relacionadas principalmente con el ordenamiento de las operaciones para llevar una réplica de un estado de procesamiento a otro Existen diversas técnicas para lograr este resultado, y se relacionan principalmente con el nivel de rigor con que se garantiza la replicación activa, y el impacto en el desempeño para el sistema en replicación.

Mientras más riguroso es un esquema de replicación, más degradado se verá el desempeño del sistema en alta disponibilidad.

Además de proporcionarse un vehículo para la transferencia ordenada de operaciones, los sistemas en Replicación Activa requieren mecanismos para soportar las nociones de grupos de replicas con servicios de membresía que identifiquen, en un sentido de distribución que procesos son miembros de un grupo en un momento dado del tiempo Se requieren mecanismos adicionales que permitan a los miembros unirse a un grupo, atrapar el estado de procesamiento del mismo, y removerse ya sea en forma voluntaria o como respuesta a una falla detectada en un miembro De aquí, los sistemas de Replicación Activa incluyen la detección de falla y

mecanismos heartbeat. Un mecanismo heartbeat es un método para estar censando un dispositivo y detectar si se pierde señal del mismo, lo que implica que ha fallado.

Los siguientes términos, son en su fondo sinónimos pero sus nombres hacen énfasis en los distintos aspectos de un mismo comportamiento:

Acrónimo	Término	Descripción
HS	Hot Standby	Acentúa la asociatividad de este modelo con otros métodos de Alta Disponibilidad que son menos exigentes. Un término mal aplicado, pues standby (reserva/sustituto) implica asimetría, siendo que este modelo especifica que debe ser replicación simétrica.
AR	Active Replication	Hace énfasis en su referencia al servidor, al indicar que la replicación que debe ocurrir requiere controlar el estado de procesamiento del servidor para lograr este objetivo. Este término es más descriptivo del mecanismo utilizado, que de la función que se espera que realice este mecanismo.
VS	Virtual Synchrony	Un término que acentúa el paradigma computacional general alcanzado cuando los grupos de procesos han garantizado el ordenamiento de mensajes entre ellos. La sincronización virtual enfatiza que el ordenamiento de procesamiento y la replicación son técnicas generales, donde la Alta Disponibilidad es sólo un (muy importante) uso de las mismas.
MFP	Make Forward Progress	Acentúa la visión de parte del cliente Se enfatiza más en la función esperada del sistema de Alta Disponibilidad, no en la forma en que se implementan los mecanismos de HA.

Tabla 4. Algunos términos sinónimos de replicación activa

2.1.6.8 La Replicación Pasiva

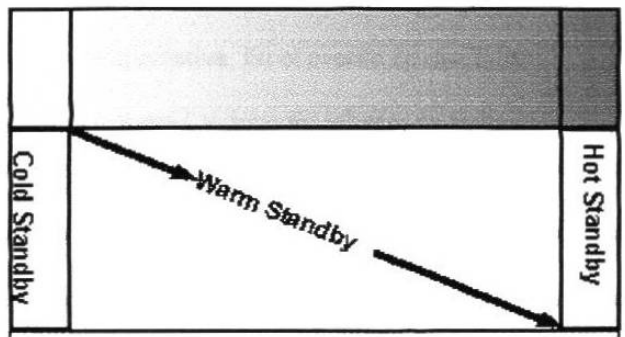


Figura 5. Espectro de Variación de modelos de alta disponibilidad, las soluciones Cold Standby y Hot Standby son puntos extremos, y en un amplio rango habrían implementaciones de tipo Warm Standby.

Describe el conjunto de técnicas usadas para lograr un modelo de Reserva Parcial (Warm Standby). Según la definición de WS, un sistema replicado con cierto nivel de inicialización y compartimiento del estado de procesamiento de la copia maestra con la réplica. En la práctica, las soluciones WS se usan como una forma relajada (menos rigurosa) de Reserva en Caliente para Alta Disponibilidad, puesto que la Replicación Activa puede ser muy costosa en términos de Performance y complejidad del diseño de la aplicación El nivel de relajamiento de los rigurosos modelos de Replicación Activa / reserva en Caliente (HS/AR), sin embargo, provoca que muchas implementaciones se definen a sí mismas como WS (Warm Standby), pero tienen una variación muy amplia entre los niveles de inicialización y estado compartido.

Algunos ejemplos válidos de Warm Standby son:

- Técnicas comunes de Replicación que se usan en algunas bases de datos, en las cuales una imagen maestra periódicamente (cada minuto/hora/noche) envía un log de transacciones a la imagen replicada, y aquí se aplican estos cambios Estas replicas pueden estar desincronizadas por el ultimo número de registros desde la última transferencia

- La imagen de procesos en memoria que se instancían (se activan) en los procesadores tanto activo como en reserva (espera) El proceso maestro administra todos los clientes que se conectan, mientras el otro procesador se mantiene a la expectativa. En el evento de una falla, todos los clientes se reconectan al proceso en reserva (espera), y deberán rehacer su trabajo desde que empezaron a trabajar en su último proceso.

- La imagen de procesos en memoria que se instancían en los procesadores activos y de espera. El proceso maestro atiende los requerimientos de los clientes. Cuando el cliente indica cerrar la transacción (hacer commit), el proceso maestro envía un log (registro) de transacciones al proceso en reserva o en espera. En el evento de una falla, los clientes son desconectados y deberán reconectarse, una vez reconectados, están en un punto de sincronización igual a la última actividad realizada, y no será necesario reiniciar todo el proceso (excepto la última actividad desarrollada)

Existen muchas otras implementaciones de replicación que podemos calificar como Warm Standby. El término LR (Lazy Replication) frecuentemente se asocia al concepto "Rollback&Recover(RR)" –Deshacer y Recuperar Puesto que con ciertos mantenimientos del estado de procesamiento, los clientes se interrumpen cuando hay un evento de falla, y deberán reconectarse explícitamente a la réplica en reserva La siguiente tabla resume estos términos.

Acrónimo	Término	Descripción
WS	Warm Standby	Se centra en la parte media del espectro de HA, y el grado variable de inicialización del estado de proceso de la replica de reserva Termino que implica el compromiso entre las alternativas Hot Standby y Cold Standby
LR	Lazy Replicati on	Se centra en la vista desde el servidor, identificando la replicación que debe ocurrir en el estado de proceso del servidor para lograr el objetivo. Termino descriptivo del mecanismo de replicación, no de la función realizada por este

		mecanismo.
RR	Rollback & Recover	Se centra en la vista desde el lado del cliente. Término descriptivo del comportamiento de un cliente en un sistema en replicación parcial (Warm Standby), por ejemplo: Los efectos visibles y trastocados de una falla. Pero no define los mecanismos de implementación.

Tabla 5: Algunos sinónimos de replicación pasiva

2.1.6.9 El Balanceo de Cargas y la Alta Disponibilidad

Mecanismo usado para lograr escalabilidad, al distribuir el procesamiento y el trabajo a través de un pool de servidores De hecho, este término no tiene nada que ver con la replicación y la alta disponibilidad. En la práctica, sin embargo, estos conceptos se asocian frecuentemente a HA debido a la inversión hecha en la adquisición de sistemas redundantes para HA que no podrían justificarse si el equipo adicional se quedara inactivo, o simplemente duplicando el trabajo realizado en los servidores primarios En vez de esto, un requerimiento frecuente de los clientes es tener un pool de servidores replicados que estén preparados para sustituir a otro en el evento de una falla (rol de Alta Disponibilidad), pero que además se dividan la carga de los clientes en condiciones normales de operación (rol de Balanceo de Carga). Tal requerimiento, aunque común, puede complicar significativamente el diseño de ambos aspectos (HA y Balanceo de Cargas), puesto que la configuración computacional no puede optimizarse específicamente para un rol

En suma, deben considerarse las implicaciones funcionales de tan comprometedores diseños Si un sistema se afina para soportar una carga normal de clientes con un pool de N servidores para balanceo de cargas, y llega a sufrir la falla de un servidor, la carga debe ahora redistribuirse entre los N-1 servidores Esto claramente impactará el desempeño global del sistema, y por lo tanto no enmascarará apropiadamente la falla (pues se verá una degradación de desempeño) Por

supuesto, el sistema puede haber sido sobreprotegido con más de N servidores, lo cual permite que ocurra un número M de fallas antes de impactar al desempeño. Sin embargo, esto nos regresa al punto de partida de adquirir capacidad adicional que estará inactiva (en condiciones normales) En la mayoría de los casos, se usa N = 2, y se define el requerimiento de Balanceo de Cargas para distribuirse en ambos equipos El añadir un tercer servidor para protección no puede ser una opción.

Aquí podemos usar una analogía Consideremos una póliza de seguro de vida, con una cartera de beneficios acumulativos usables en vida. El propósito inicial de la póliza es proteger a la familia del asegurado en el evento de fallecimiento o catástrofe similar Sin embargo, la cartera de beneficios en la póliza puede usarse a discreción del asegurado, para solventar gastos normales de la vida diaria El peligro, por supuesto, es que al usar los recursos de la póliza durante la aseguranza en vida, sus beneficios pueden haberse reducido cuando realmente se requieren, al faltar el asegurado-

La Alta Disponibilidad (HA) y el diseño de software, puede pensarse como una póliza de seguros que tiene un balance de beneficios usables en vida. Por Ejemplo, la capacidad de procesamiento de los servidores redundantes. Al usar esta capacidad para balancear la carga durante operaciones normales, el beneficio de la HA se reduce al presentarse una falla.

2.1.6.10 La Disponibilidad y los Costos

2.1.6.10.1 El Costo de tener Alta Disponibilidad

La Disponibilidad Continua, y la Replicación Activa son un concepto agradable, sin embargo, el lograr aplicarlo conlleva un costo monetario, de desempeño y de complejidad que deberán

quedar claros si se desea implementar Necesitamos entender los requerimientos centrales de disponibilidad que deberá ofrecer el software del sistema, para poder entender el costo que tendrá el poder soportar el cumplimiento de estos requerimientos.

Es muy típico para una empresa que esta diseñando un sistema en alta disponibilidad, que los clientes del sistema definan sus requerimientos como 7x24, al cuestionarlos, siempre pedirán que "el sistema este operando a la hora que se requiera, es decir, debe ser 7x24", como si esto lo dijera todo, "Ningún dato se deberá perder nunca, y el sistema deberán permanecer activo y no se deberá presentar ninguna sensación perceptible de falla, los mantenimientos y actualizaciones de los sistemas no deberán interferir con el servicio y operación", esto es en verdad una gran expectativa.

Cuando no se está apropiadamente informado de los costos totales de adquisición, mantenimiento, e impuestos que implica el adquirir un Rolls Royce, es natural desear precisamente uno. Sólo que al darse cuenta que además del costo de adquisición, existe un costo adicional por el seguro, consumo de gasolina, partes de refacción, y costo del servicio calificado, entonces se tiene la suficiente información para decidir que opción elegir del rango de los modelos desde lujo hasta económico.

Igual que al evaluar productos de consumo final y automóviles, al evaluar las inversiones en sistemas complejos, es importante poder tener esta perspicacia y mente crítica que permita distinguir entre las nubes publicistas que pretenden vender productos como adecuadas soluciones para tareas que realmente no son capaces de hacer. Así también en el mercado de Alta Disponibilidad, existen una cantidad inmensa de folletos y mercadotecnia que se promocionan como "totalmente 7x24". Ni que decir, entonces, que los usuarios crean que este requerimiento es algo simple y sencillo que el sistema debe proporcionar

En suma, tanto los usuarios de sistemas computacionales, así como los desarrolladores y creadores de los mismos tienden a estar más interesados en especificar los requerimientos funcionales que debe tener: mas no que capacidades útiles realmente posee. Los temas de desempeño, escalabilidad, confiabilidad, etc. tienden a ser decididas en forma menos enfocada. Cuando muchas veces son estos requerimientos no asociados a la funcionalidad tan importantes como la utilidad global del sistema, pero si implican una complejidad igual o mayor que los requerimientos funcionales del sistema

Finalmente, es responsabilidad de los usuarios y los desarrolladores, el trabajar en equipo para lograr identificar claramente: (1) ¿Qué es realmente lo que requieren los usuarios? Contra lo que desean, (2) ¿Qué alternativas tecnológicas podemos usar para cumplir esas expectativas?, (3) Los costos monetarios, complejidad, algoritmos de replicación, protocolos para pertenecer a un grupo, degradación del desempeño causado por la solución.

2.1.6.10.2 El Costo que implica el NO TENER una Alta Disponibilidad

La única forma de convencernos de la importancia de la Alta Disponibilidad, es demostrando los costos que implica la ocurrencia de una falla y el efecto del downtime (o corte de servicio), pero desde una perspectiva de dólares y centavos

El costo más obvio de un downtime, no es necesariamente el costo más caro o impactante de todos Uno de los costos más obvios del downtime es que el usuario pierde productividad, pero el costo real depende de que tanto trabajo esta dejando de realizar el usuario del sistema afectado, y un costo aun mayor es: ¿Está afectando esto a la imagen de mi compañía?

Si los usuarios de un sistema fallado son desarrolladores, el costo puede no ser muy impactante en una empresa usuaria, pero si es una empresa de software, este costo si será grande Si un desarrollador tiene un costo de entre $400 a $1000 el día, es muy razonable que un grupo de 50 desarrolladores inactivos causarían un elevado costo de hasta $2,000,000 en la semana Si se trabaja por proyecto y existe un entregable, además de este costo, deberemos agregar las horas extras para poder salir a tiempo, si se retrasa el proyecto, entonces deberemos agregar las penalizaciones, además de un costo intangible que es la imagen.

En sistemas operacionales, el costo por un downtime puede identificarse por el costo por hora de los usuarios inactivos de los sistemas afectados, sin embargo, habrá que agregar los pedidos perdidos, las ventas no hechas, las inversiones no efectuadas, reaprovisionamiento de inventarios no realizados, etc. Otros costos son la pérdida de imagen ante clientes potenciales (principalmente en el comercio electrónico), o los nominados costos de oportunidad Imaginemos una firma de la casa de bolsa que no pudo realizar la puesta de acciones en el momento antes de que bajaran de precio, también pudo ocurrir que haya evitado vender acciones que subían de precio (ahorrándole dinero a la firma)

Muchos costos no son cuantificables, pero tratemos de poner una ejemplo Imaginemos que queremos comprar un CD de música o un libro en un negocio de ventas vía Internet, entramos al sitio y vemos la descripción del producto, pero cuando queremos levantar el pedido, nos responde con un mensaje indicando que en este momento el sistema no está operando, intente más tarde, pero yo quiero pedir ya este producto Entonces, busco otro sitio de Internet (ya se perdió una venta, pero además ayude a mi competencia), si además en este negocio me entregan el producto rápida y eficientemente, cuando quiera comprar otra cosa ¿donde lo pediré? Definitivamente en el segundo negocio (más pérdidas) Imaginemos que un amigo me dice, oye yo quería conseguir ese disco, y los centros comerciales no le he visto, seguramente le diré, mira

en Internet busca este sitio y ahí lo encontrarás (otra pérdida más), y si comentamos que en el primer sitio no es muy confiable ¿quién ganará más ventas? Y si esto lo platicamos en una reunión de amigos, ¿Cuantos clientes se han perdido?

2.1.6.10.3 Cuando es más impactante el costo de la disponibilidad.

La diferencia de efecto entre los cortes de servicio de un sistema interno a una empresa, y los cortes de servicio en un sistema de Internet es que mientras la falla de un sistema interno de una empresa puede ser cubierto por los empleados, la falla de un sistema externo vía Internet, impacta instantáneamente a los clientes y no puede ser cubierta la sensación de falla

El corte de servicio debe observarse desde la perspectiva del usuario, e incluye la inhabilidad para acceder o comprar productos o servicios desde él site (por cualquier razón), así como la aparición de problemas de desempeño

El desempeño es un atributo de la disponibilidad, y cuando el tiempo de respuesta toma más de lo que un usuario esta dispuesto a tolerar (tipicamente un valor entre 6 a 10 segundos), estos habrán de considerar que los servicios de la empresa están fuera de operación y le abandonará.

En la nueva economía de redes, el costo de corte de servicio es usualmente mucho mayor que en los ambientes comerciales físicamente localizados Las pérdidas por corte de servicio, afectan a las ganancias reales (ventas perdidas durante los períodos de falla, que no se habrán de recuperar posteriormente), las posibles ganancias de clientes actuales y prospectos de cliente, debido a la publicidad negativa, y daños irreparables a la reputación empresarial La complejidad de la mayoría de las infraestructuras de aplicaciones Web, también dificulta el asegurar la consistencia de los servicios el 100% del tiempo operacional. Más aún, los costos de obtener mayores niveles

de disponibilidad se incrementa en forma exponencial. Para poder sostener y justificar las inversiones en disponibilidad, las empresas deben calcular un retorno de la inversión que garantice que el beneficio (la reducción de los costos por cortes de servicio) excede el costo mismo de la inversión.

2.1.7 Diferencia entre Fault-Tolerance y Alta Disponibilidad

El objetivo principal de un sistema Fault-Tolerance es tener la máxima exactitud de la información procesada durante el tiempo que el sistema este operando. Un sistema Fault-Tolerance no siempre esta disponible pero un sistema Fault-Tolerance siempre deberá ser exacto (recordemos los cajeros automáticos de los bancos, que hacen cortes de servicio cada cierto tiempo, pero no deben fallar al momento de realizar una transacción)

Un sistema en Alta Disponibilidad estará casi siempre en servicio, pero no siempre se requiere un procesamiento con nivel extremo de exactitud: por ejemplo, un Proveedor de Servicios de Internet siempre debe estar operando, aun cuando en ocasiones las imágenes que presenta el site se vean indefinidas o borrosas Otro caso es el de las consultas telefónicas de saldos, usualmente este tipo de servicio debe ser 7x24 aun cuando la información no siempre es exacta, pues el saldo será proporcionado a una fecha de corte (sin incluir los últimos movimientos de la cuenta).

La definición de un sistema Fault-Tolerance consiste de equipo y sistemas con tecnología propietaria de alto costo, y sistemas duplicados fuertemente acoplados. El manejo de fallas se integran y se convierten en parte de las funciones del sistema operativo. Estos sistemas tienen una respuesta automática y espontánea a las fallas de sistema y proporciona servicios continuos ininterrumpidamente

2.2 Esquemas de Alta Disponibilidad

Para todo negocio (principalmente si es de comercio electrónico) es impactante el nivel de competitividad que le da el tener un sistema siempre disponible, el cual permite atender los pedidos, y llamadas de clientes potenciales que al recibir un servicio eficiente (y a cualquier hora) se pueden convertir en clientes leales.

Definitivamente la alta disponibilidad tiene un costo, pero siempre será un costo que se estará dispuesto a pagar si logramos este nivel de servicio y si logramos garantizar la respuesta y funcionalidad del sistema aún en los momentos más críticos de operación (las horas pico)

El impacto más fuerte que puede tener una empresa al no usar esquemas de alta disponibilidad es bajo el rubro de imagen, ya que se pone en entredicho la credibilidad de una empresa que no puede garantizar el nivel servicio de los sistemas de primer contacto con el cliente.

2.2.1 Clustering

El concepto de cluster es tomar dos o más computadoras independientes(recién desempacadas), y organizarlas para trabajar en conjunto para proporcionar la más alta disponibilidad y escalabilidad que puede obtenerse al usar un solo sistema Cuando una falla ocurre en un cluster, los recursos pueden ser movidos a otro sistema en el cluster y la *recuperación* del trabajo perdido es posible mediante procedimientos de software. Al contrario, un sistema tolerante a fallas usa un hardware especial para correr múltiples computadoras en un modo *"paso a paso"* tal que se proporcione un servicio de computo que no se detenga al ocurrir una falla de algún componente. Los sistemas tolerantes a fallas son mucho más caros debido a el hardware de propósito especial que necesitan.

2.2.1.1 Beneficios de los clusters.

La arquitectura de cluster puede proporcionar tres principales beneficios hacia el usuario:

- **Mejora la Disponibilidad:** Al proporcionar un servicio continuo aún durante una falla de hardware o software. Cuando el servidor falla, la carga de trabajo es reasignada a los servidores restantes, la sensación es de un corto período sin servicio, tan corto que la expectativa es que no sea perceptible, excepto por aquellas transacciones que estaban en proceso justo en ese momento. El proceso de failover depende de la aplicación en uso.
- **Mejora la Escalabilidad:** Al permitir que se agreguen nuevos componentes conforme la carga del sistema se incrementa. Cuando la carga global excede las capacidades de los equipos en el cluster, se pueden agregar nuevos módulos (para esto el sistema del cluster debe ser capaz de redistribuirse en varios equipos a la vez). Se puede incrementar el poder de procesamiento al agregar más CPU's (un equipo a la vez). Adicionalmente, se puede incrementar aún más la capacidad de procesamiento al aumentar el número de servidores en el cluster
- **Una Administración Simplificada:** En los grupos de sistemas y sus aplicaciones, al permitir que los administradores gestionen los grupos completos de servidores y procesos como un solo sistema.

2.2.1.2 Limítrofes de la tecnología de clusters.

Los clusters pueden proteger contra fallas en el almacenamiento, fallas de hardware, fallas en un segmento de la red (enrutando los servicios por una red alterna), pero se requiere que el software de aplicaciones sea funcionalmente capaz de recuperarse de un evento de falla en un cluster

- **La tecnología de cluster** no puede proteger contra fallas inducidas por corrupción de software, o por fallas causadas por negligencia o descuido de la intervención humana. Si el sistema operativo del servidor se cae de tal forma que corrompe las definiciones de cluster, la recuperación de la información procesada por este miembro puede no ser recuperable por otros elementos del cluster.
- **La presencia de un virus** o una falla del software de aplicación causa que la estructura lógica de datos de la aplicación se corrompa, probablemente no será posible recuperar la aplicación mediante la operación normal del cluster, requerirá procedimientos avanzados de recuperación.
- **El borrar un archivo** con información importante en forma accidental del sistema, cuando el usuario esta operando, definitivamente no será recuperable (mediante principios del cluster), si es un archivo de cierto tiempo de vida, probablemente se tenga algún respaldo, pero requiere intervención manual Si el archivo se generó en el momento o como parte de la operación normal, entonces no existe respaldo y se puede considerar una perdida total

2.2.2 Sites Espejo

Este concepto es un nivel más avanzado de alta disponibilidad y esta enfocado a la recuperación de desastres La recuperación de desastres es el nivel máximo de alta disponibilidad, y se enfoca a tener un site remoto que es (o debe ser) un espejo de los sistemas y equipos de mayor importancia ubicados en el site principal, en caso de que este site principal llegará a sufrir una falla mayor: Un incendio, una bomba, terremoto, inundación, o caída de un edificio. El site remoto estará en posibilidad de retomar el control de las operaciones y lograr que el negocio continúe en la operación Un estudio efectuado por la Universidad de Texas acerca de las compañías que han sufrido una pérdida catastrófica de información, el 43% nunca volvió a operar, el 51% cerró en el transcurso de 2 años a partir de la fecha del desastre, y sólo 6 sobrevivieron Los sites espejo y los

planes para ponerlo en operación en caso de desastre, es decir, los Planes de Recuperación de Desastre son imperativos ya que una empresa no logrará recuperarse en caso de que no tenga establecido el suyo al momento de ocurrir un desastre mayor.

Adicionalmente, deberá existir una metodología para el traslado de cintas con los respaldos de la información desde un edificio a otro, tal que se tenga una fuente parcial de información en caso que el site remoto no logre operar adecuadamente

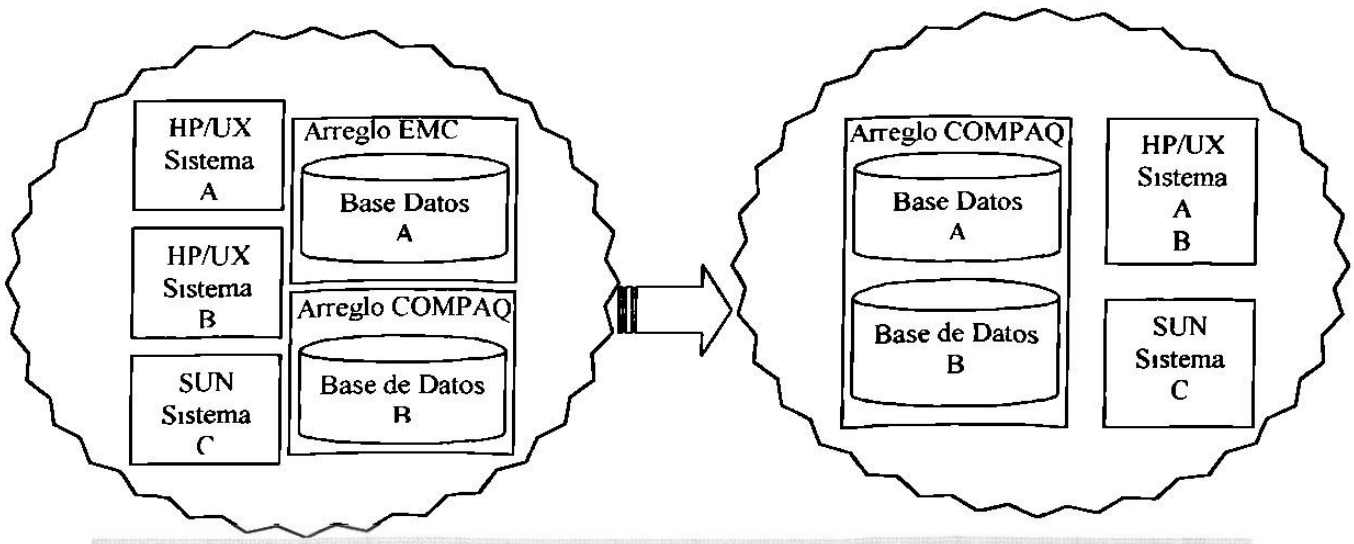
Existen diversas formas de lograr mantener la configuración de un site espejo, una es realizando una copia de cada una de las bases de datos del site principal, y posteriormente aplicando las operaciones en forma incremental. Otra forma es mediante hardware especial que proporciona replicación byte a byte de cada uno de los discos hacia un juego de discos remoto, en caso de falla, la copia de información de los discos estará actualizada hasta un momento muy cercano a la falla, con lo cual se podrá retomar la operación en el site remoto.

Un site remoto, no necesariamente debe ser una réplica exacta del site primario, pero si debe tener una copia de la información organizada de tal forma que la operación pueda ser retomada, aún cuando los equipos que se utilicen para tal efecto, no sean necesariamente de la misma capacidad que el site primario

El modelo de Recuperación de Desastres debe tener un plan de implementación, así como un plan para aplicarlo en caso de presentarse la necesidad, es un plan muy complejo, que debe contemplar desde las instalaciones eléctricas duplicadas, redes de comunicación duplicadas, arreglos de discos en replicación, equipos de computo adicionales, racks para soportar estos equipos, aire acondicionado alterno, sistema contra incendios, fuentes de energía, y si esto no fuera poco, esto

mismo deberá existir para el site remoto, con costos que deben calcularse adecuadamente y justificarse para poder lograr su aplicación.

En general, el esquema de recuperación de desastres y el manejo de Sites Espejo es un tema demasiado amplio por si mismo suficiente para dedicar una investigación propia al mismo. Por esta razón no se contempla profundizar más sobre este tema en este proyecto, sino mencionar que



Site Principal **Site Remoto**

El site remoto, no necesariamente es una copia fiel del site principal, sin embargo debe estar preparado para potestad recibir la carga y soportar la operación (tal vez un poco más lenta) pero que permita continuar las actividades del negocio.

Figura 6. Ejemplificación de un esquema de recuperación de desastres

existe y es el esquema de mayor protección que puede aplicar una empresa. Adicionalmente, ya existen algunas alternativas de solución a esta necesidad, y muchos proveedores ya son capaces de soportar esquemas de automatización para lograr eficientes implementaciones de los mismos