



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE SISTEMAS Y COMPUTACIÓN

“Proyecto de Investigación previo a la obtención del título de Ingeniero en Sistemas y
Computación”

TRABAJO DE TITULACIÓN

**IMPLEMENTACIÓN DE UN DATA CENTER VIRTUAL PARA LA
APLICACIÓN DE LA TÉCNICA TOLERANCIA A FALLOS EN SERVIDORES
USANDO TECNOLÓGICA VMWARE.**

Autor(es):

Cristhian Alfonso Cobos Cevallos

Tutor:

Ing. Diego Marcelo Reina Haro

Riobamba - Ecuador

2021

PÁGINA DE ACEPTACIÓN

Los miembros del tribunal de Graduación del proyecto de investigación de título: **“IMPLEMENTACIÓN DE UN DATA CENTER VIRTUAL PARA LA APLICACIÓN DE LA TÉCNICA TOLERANCIA A FALLOS EN SERVIDORES USANDO TECNOLÓGICA VMWARE.”**, presentado por el estudiante Sr. Cristhian Alfonso Cobos Cevallos, dirigido por el MsC. Diego Marcelo Reina Haro.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación escrito, con fines de graduación en el cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Mgs. Diego Reina

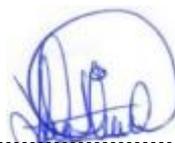
2.1. Tutora del Proyecto



Firma

Mgs. Marlon Silva

2.2. Miembro del Tribunal



Firma

Mgs. Andrés Cisneros

2.3. Miembro del Tribunal



Firmado digitalmente
por ANDRÉS
SANTIAGO CISNEROS
BARAHONA

Firma

DERECHO DE AUTORÍA

La responsabilidad del contenido de este proyecto de graduación corresponde exclusivamente a: Cristhian Alfonso Cobos Cevallos bajo la dirección del Mgs. Diego Reina, y al patrimonio intelectual de la Universidad Nacional de Chimborazo.

Autor



.....
Cristhian Alfonso Cobos Cevallos
0104591565

Director del Proyecto



.....
MsC. Diego Reina
0602973109

DEDICATORIA

El presente trabajo se lo dedico a mis padres por su apoyo incondicional y por ser artífices en la culminación de mis estudios, a mi esposa y mis hijos por ser fuente de motivación e inspiración para superarme cada día y a mi familia en general por el apoyo que siempre me han brindado a lo largo de mi carrera universitaria.

AGRADECIMIENTO

En primer lugar, quiero dar gracias a Dios por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por poner en mi camino a aquellas personas que han sido mi soporte y compañía durante toda mi carrera.

Agradezco a la Universidad Nacional De Chimborazo y a todos sus docentes por las oportunidades que me han brindado, por haberme permitido formarme como profesional y como persona de bien.

Quiero agradecer a mis padres por respaldarme en cada decisión y proyecto que he realizado dándome el mejor ejemplo de superación, humildad y sacrificio.

A mi esposa e hijos por todo el amor y comprensión durante todo este tiempo, por acompañarme y compartir conmigo todos mis triunfos y fracasos.

A mi tutor Msc. Diego Reina por su valiosa guía, asesoramiento y por enriquecer con sus conocimientos y sugerencias en el desarrollo de esta tesis.

Y gracias a todas las personas que ayudaron directa o indirectamente en la realización de este trabajo.

ÍNDICE GENERAL

DERECHO DE AUTORÍA	III
DEDICATORIA.....	IV
AGRADECIMIENTO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	VIII
RESUMEN.....	IX
ABSTRACT	X
INTRODUCCION.....	1
CAPITULO I.....	3
PLANTEAMIENTO DEL PROBLEMA.....	3
1.1. Problema y justificación de la investigación	3
1.2. OBJETIVOS.....	4
1.2.1. Objetivo General	4
1.2.2. Objetivo específico.....	4
CAPÍTULO II.....	5
MARCO TEÓRICO	5
2.1. Virtualización de Redes y Servidores Emulando Infraestructuras Tecnológicas..	5
2.2. Historia de la virtualización	5
2.3. Ventajas del Datacenter virtualizado	6
2.4. Definición de virtualización.....	7
2.4.1. Ventajas de la virtualización.....	7
2.5. Hipervisores	7
2.5.1. VMware Workstation	8
2.5.2. VMware ESXi	8
2.5.3. VMware vSphere.....	8

2.5.4.	VMware vCenter Server	9
2.5.5.	OpenFiler	10
2.5.6.	vSphere vMotion	10
2.6.	Clasificación de los Clúster	10
2.6.1.	Alta disponibilidad	10
2.6.2.	Alto rendimiento.....	10
2.7.	ADAD-SW.....	11
2.8.	Virtualización de Redes y Servidores Emulando Infraestructuras Tecnológicas	11
2.9.	Requerimiento de redundancia	12
2.10.	Requerimiento de protección y confiabilidad	12
2.11.	Requerimiento de implementación de clúster	12
2.12.	Requerimientos de replicación	13
2.13.	Modelos de Virtualización	13
2.14.	Modelo de Máquina Virtual	13
2.15.	Modelo de Máquina Paravirtual.....	13
2.16.	Modelo de Virtualización a nivel de Sistema Operativo.....	14
2.17.	Tolerancia a Fallos para Infraestructura TI.....	14
2.17.1.	Ventajas	18
2.17.2.	Desventajas	19
2.17.3.	Software	20
	CONCLUSIONES.....	32
	RECOMENDACIONES	33
	BIBLIOGRAFÍA	34

ÍNDICE DE FIGURAS

Figura 1: Infraestructura de VMware	8
Figura 2: Infraestructura vSphere client	9
Figura 3: Infraestructura vCenter Server	9
Figura 4: Clasificación de los Clúster	11
Figura 5: Requerimientos	14
Figura 6: Garantía de funcionamiento.	16
Figura 7: Niveles de error.	17
Figura 8: Migración de máquinas virtuales	25
Figura 9: Selección de servidor	25
Figura 10: Conformación del servidor espejo.	26
Figura 11: Tiempo de conformación del servidor espejo.	26
Figura 12: Máquina Windows 10 aplicada técnica tolerancia a fallos.	27
Figura 13: Espejo creado en el segundo servidor.	27
Figura 14: Máquinas virtuales primer servidor.	27
Figura 15: Máquinas virtuales segundo servidor.	28
Figura 16: Modo mantenimiento servidor 1.	28
Figura 17: Migración máquinas virtuales servidor 1.	29
Figura 18: Máquinas virtuales alojadas en el servidor 2 después de modo mantenimiento servidor 1.	29
Figura 19: Modo mantenimiento servidor 2.	30
Figura 20: Máquina activada la tolerancia a fallos en servidor 2.	30
Figura 21: Espejo de Ubuntu 14 s1 en servidor 2.	31

ÍNDICE DE TABLAS

Tabla 1: Garantía de Rendimiento	15
Tabla 1: Matriz de consistencia	23

RESUMEN

El presente proyecto de investigación tiene como objetivo implementar la técnica de tolerancia a fallos usando tecnología VMware para garantizar la alta disponibilidad de los servidores, mediante una infraestructura virtual, convirtiéndose en una alternativa de cualquier PYMEs, en la actualidad la virtualización se ha convertido en una herramienta muy eficiente debido a la capacidad para conformar sistemas de software de varios servidores en uno solo. Con este método se reduce significativamente el costo de implementación y mantenimiento de hardware, mejora la productividad, eficiencia y sobre todo se logra una alta disponibilidad.

Para el desarrollo del proyecto se realizó la creación de la infraestructura virtual en VMware vSphere 5.1 el cual tiene un clúster con dos servidores ESXi 5.1, los cuales constan de 2 máquinas virtuales cada uno, conectados a un sistema de almacenamiento Openfiler, Bajo VMware Workstation 15.

La técnica de tolerancia a fallos permitirá garantizar la alta disponibilidad de los servicios alojados en los servidores ESXi, así como se reducirá el costo de ampliación de recursos ya que no se tendrá que adquirir equipos nuevos. Por otra parte, Vmotion nos permite una migración dinámica de un servidor a otro sin tiempo de inactividad para cargas de trabajo. Finalmente se pudo lograr la alta disponibilidad de los servicios mediante la técnica de tolerancia a fallos en un entorno virtualizado.

Palabras clave: Virtualización, Servidores ESXi, vCenter server, vSphere 5.5, fault tolerance, Vmotion.

ABSTRACT

This investigation project aims to implement the tolerance technique by fault using VMware technology to guarantee the high availability of servers through a virtual infrastructure, becoming an alternative for any PYMEs. Virtualization has become a very efficient tool due to the ability to form software systems of several servers in one. With this method, hardware implementation and maintenance costs are significantly reduced, productivity and efficiency are improved, and above all, high availability is achieved.

For the project's development, the virtual infrastructure was created by VMware vSphere 5.1, a cluster with two ESXi 5.1 servers. They consist of 2 virtual machines each, connected to an Openfiler storage system, under VMware Workstation 15.

The tolerance technique by fault will permit the high availability of the services hosted in the ESXi servers and reduce the cost of expanding resources because no new equipment will have to be purchased. On the other hand, Vmotion allows us a dynamic migration from one server to another without losing time for workloads.

Finally, it was possible to achieve the high availability of the services using the tolerance technique by a fault in a virtualized environment.

Keywords: Virtualization, ESXi Servers, vCenter server, vSphere 5.5, fault tolerance, Vmotion.

Reviewed by:
Ms.C. Ana Maldonado León
ENGLISH PROFESSOR
C.I.0601975980

INTRODUCCIÓN

La tecnología de virtualización desarrollada a finales de los sesenta reapareció a finales de los noventa. Esto se debe a los cambios en la forma en que se procesa la información, los avances en el hardware y las características atractivas proporcionadas por la tecnología, como la capacidad de integrar múltiples servidores virtuales en la misma computadora física y el contenedor de aislamiento para ejecutar aplicaciones. Seguridad y protección (Pessolani et al., 2012).

La virtualización ha sido identificada como una de las diez principales tecnologías estratégicas. Se trata de extraer software de un ordenador y encapsularlo en lo que llamamos máquina virtual, que se ejecutará en una máquina física distinta a la anterior. Después de ser aplicado a la red de servicios internos de la organización, cambiamos exitosamente de una Intranet con algunos servidores infrautilizados a una Intranet con una pequeña cantidad de servidores que brindan el mismo servicio (Martín, Marrero, Urbano, Barra, & José-Antonio, 2011).

Debido a que la empresa ha utilizado la tecnología de la información y las comunicaciones para mejorar sus procesos, ha creado miles de centros de datos que contienen todos los componentes de hardware que dan soporte a las aplicaciones de gestión de la información. Algunos de estos componentes son componentes de red, que se encargan de mantener la intercomunicación entre todos los puntos de acceso de información, y otros componentes son componentes de almacenamiento que se utilizan para almacenar todos los datos obtenidos de las aplicaciones, y además incluyen al servidor como eje central. Se ejecutará todo el procesamiento de la información en la red de la organización (Galán, 2015).

En general, los centros de datos tienden a crecer a medida que crece la demanda de información porque es necesario agregar componentes para apoyar los servicios, las aplicaciones y la información. Cuando esto sucede, se necesitan más y más energía y espacio físico para acomodarlos. Tendencia general Para que se brinden aplicaciones o servicios de TI, debe haber un servidor que lo soporte, aunque en algunos casos, un cierto número de estos servicios o aplicaciones se pueden agrupar en un solo servidor. A En última instancia, esto resultó en altos costos debido a varios factores: energía, mantenimiento, administración y compra de otros componentes (Galán, 2015).

La principal motivación es aumentar el uso compartido y el uso de recursos informáticos costosos (como los mainframes). La virtualización es un método lógico para dividir las computadoras centrales en máquinas virtuales independientes. Estas particiones permiten al mainframe realizar tareas múltiples, es decir, ejecutar muchas aplicaciones y procesos al mismo tiempo. Dado que los mainframes eran recursos costosos en ese momento, se diseñaron para poder dividirlos y aprovechar al máximo la inversión (vmware, www.vmware.com, 2019).

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. Problema y justificación de la investigación

Considerando que uno de los factores más importantes en la creación de un centro de datos es asegurar la continuidad de los servicios brindados a clientes, empleados, ciudadanos, proveedores y empresas colaboradoras, por lo que en la mayoría de las empresas del mundo ya cuentan con la infraestructura de los datos centro. En estos campos, la protección física de los equipos informáticos o de comunicaciones involucrados y el servidor de la base de datos que puede contener información crítica son muy importantes. Del mismo modo, el crecimiento de la información a gestionar también ha provocado el crecimiento de la infraestructura necesaria para almacenar, mantener y gestionar la cantidad de datos, lo que plantea diversos retos para la gestión de los centros de datos de diferentes empresas en todo el mundo.

La ampliación de los centros de datos o su mejoramiento requiere de adquisición de nuevos recursos y mano de obra por lo cual resulta muy costoso y demorado de implementar, es por esta razón que se está buscando nuevos métodos para mejorar los centros de datos y la disponibilidad de ellos.

Con la implementación de la técnica de tolerancia a fallos en servidores virtualizados con tecnología VMware se podrá evidenciar una alternativa para muchas empresas que desean mejorar su disponibilidad sin la necesidad de adquisición de nuevos centros de datos que resultan muy costosos y altos en consumo de energía, por lo tanto esta alternativa brindara una manera más eficiente de operabilidad y disponibilidad.

El proyecto de investigación es viable, porque gracias a la técnica de tolerancia a fallos configurado en un data center virtual, mejora la disponibilidad de los servicios alojados en dicho data center con un costo más bajo.

Al configurar la técnica de tolerancia a fallos en un data center virtual se logra tener una alta disponibilidad de los servicios alojados con un costo muy inferior.

1.2. OBJETIVOS

1.2.1. Objetivo General

- Implementar la técnica de tolerancia a fallos para servidores que forman parte del clúster de un data center apoyado en la tecnología VMware.

1.2.2. Objetivo específico

- Realizar un estudio sobre la técnica de Tolerancia a Fallos para infraestructuras TI.
- Implementar mediante virtualización una infraestructura tecnológica TI o Data Center con clúster de servidores utilizando VMware para la configuración de la técnica de Tolerancia a fallos.
- Evaluar la aplicación de Tolerancia a fallos en un escenario de pruebas.

CAPÍTULO II

MARCO TEÓRICO

2.1. Virtualización de Redes y Servidores Emulando Infraestructuras Tecnológicas

La virtualización es una tecnología con un gran potencial, que le permite administrar de manera efectiva el hardware, el software, la integración del servidor, el costo, el espacio físico y los recursos humanos en la infraestructura de TI, al tiempo que mejora las capacidades de administración. Y seguridad de escritorio virtual (Vilac, 2012).

Actualmente, debido a sus poderosas capacidades de almacenamiento de información y la capacidad de manejar aplicaciones y servicios a gran escala, la implementación de centros de datos en la compañía ha aumentado para optimizar los costos operativos de la infraestructura técnica. A partir del centro de datos, hoy ha surgido la llamada virtualización de servidores, que permite instanciar múltiples máquinas virtuales en una sola máquina física (Vilac, 2012).

La zona desmilitarizada (DMZ) contiene un conjunto de patrones de diseño de red probados que pueden trabajar juntos para resolver estos problemas para los científicos. Presentamos el modelo Science DMZ, que incluye arquitectura de red, configuración de sistema, seguridad de red y herramientas de rendimiento, que crean un entorno de red científicamente optimizado (Dart, 2014).

El modelo de zona dinámica desmilitarizada (DMZ) tiene en cuenta tanto el rendimiento de la red como la seguridad. Y responde dinámicamente a las demandas de tráfico en tiempo real (Dart, 2014).

2.2. Historia de la virtualización

En la década de 1960, los equipos de TI de muchas empresas y agencias gubernamentales encontraron problemas similares: tenían supercomputadoras o "mainframes" de alto rendimiento y querían "dividirlos lógicamente" o usarlos para múltiples tareas simultáneas. Es por eso que IBM desarrolló un método para crear múltiples "particiones lógicas" (similar a lo que hoy llamamos "máquinas virtuales"), que funcionan independientemente unas de otras, y todos usan los recursos proporcionados por el "mainframe" (Lemus, 2020).

En la década de 1980, con el advenimiento de las máquinas x86 relativamente económicas, había comenzado una nueva era de microcomputadoras, aplicaciones cliente-servidor y "computación distribuida". En una caja enorme y cara, un "mainframe" grande y potente con mil una tareas y utilidades ocupa mucho espacio en la sala de servidores. Entonces, por ejemplo, es mucho más económico usar un mainframe para asignar recursos para tres tareas en lugar de tener tres computadoras cada una realizando una tarea separada. Pocas personas participaron en la virtualización en ese momento, e incluso se puede decir que se olvidó. A finales de los 90, debido al desarrollo del hardware, reapareció un problema muy similar al de los 60: el hardware existente era muy eficiente y fácil de usar. Cada "caja" de una sola aplicación desperdiciará recursos, espacio, energía y dinero (Lemus, 2020).

Por lo tanto, ha surgido la idea de volver a particionar el hardware para que pueda usarse como varios servidores independientes, pero compartiendo los recursos del mismo servidor físico. De aquí nació lo que hoy llamamos "virtualización" (Lemus, 2020).

2.3. Ventajas del Datacenter virtualizado

Según (Rodríguez, 2013) las ventajas son las siguientes:

- Escalabilidad
- Versatilidad
- Duplicidad
- Balanceo de la carga
- Servicio ininterrumpido
- Ahorro
- Creación de entornos de pruebas

La informática de investigación se denomina informática de alto rendimiento y utiliza potentes herramientas y procesos informáticos para generar datos en la investigación académica avanzada. Con los clústeres informáticos de alto rendimiento, los centros de investigación pueden obtener la velocidad y la funcionalidad de las supercomputadoras costosas a una fracción del costo y reducir el riesgo de tiempo de inactividad prolongado (Chuquiguanca, 2015).

La reducción en el número de servidores significa que, al reducir las tareas de administración, es posible reducir los costos en términos de comprar nuevos servidores y mantenerlos. Además, la reducción en el número de servidores conduce a la optimización

del espacio físico y a la reducción del consumo de energía en términos de consumo y enfriamiento del servidor. Por lo tanto, las soluciones IaaS brindan a las empresas la posibilidad de lograr un crecimiento más sostenible en términos de economía y medio ambiente (Guirao, 2011).

2.4. Definición de virtualización

La virtualización es una tecnología que puede crear uno o más entornos virtuales o simulados a partir de centros de datos, clústeres de servidores y máquinas virtuales de forma dedicada. La virtualización también se puede aplicar a servidores, aplicaciones, almacenamiento y redes individuales. Esto significa que, con la virtualización, el software puede utilizarla para simular la existencia de una sola pieza de hardware, con el fin de maximizar el uso de los recursos de hardware disponibles (RedHat, 2018).

2.4.1. Ventajas de la virtualización

Ventajas de la virtualización La virtualización puede mejorar la agilidad, la flexibilidad y la escalabilidad de la infraestructura de TI, al tiempo que ahorra muchos costos. Algunos de los beneficios de la virtualización, como el aumento de la movilidad de la carga de trabajo, la mejora del rendimiento y la disponibilidad de recursos o la automatización de las operaciones, simplifican la gestión de la infraestructura de TI y reducen los costes operativos y de propiedad. Otras ventajas son:

- Reducir la inversión de capital y los gastos operativos
- Reducir o eliminar el tiempo de inactividad
- Mejore la productividad, la eficiencia, la agilidad y la capacidad de respuesta de TI
Asigne aplicaciones y recursos más rápido
- Capacidades de recuperación ante desastres y continuidad empresarial mejoradas
- Gestión simplificada del centro de datos La disponibilidad de un verdadero centro de datos definido por software

2.5. Hipervisores

Los hipervisores tienen muchas ventajas, pero lo primero que hay que entender es que, si bien las máquinas virtuales o las máquinas virtuales pueden ejecutarse fácilmente en el mismo hardware físico, siguen estando separadas entre sí. En otras palabras, si, por ejemplo, una máquina virtual experimenta un bloqueo, un ataque de virus o un bloqueo, solo esa máquina virtual se verá afectada y no las demás. Por otro lado, una de las grandes

ventajas de los hipervisores es que las máquinas virtuales son móviles, por lo que se pueden migrar fácilmente o migrar a otros servidores, ya que son independientes del hardware, que en realidad es mucho más simple que las aplicaciones tradicionales vinculadas a la física; hardware.

2.5.1. VMware Workstation

VMware Workstation es una serie de productos de hipervisor de escritorio que permiten a los usuarios ejecutar clústeres, contenedores y máquinas virtuales de Kubernetes (VMWARE, 2021).

2.5.2. VMware ESXi

También conocido como VMware ESXi Hypervisor, es básicamente un sistema operativo liviano con una función de virtualización en su núcleo. Esto significa que es un sistema operativo de hipervisor. VMware ESXi se instala directamente en el servidor físico, por lo que podemos crear múltiples servidores lógicos o máquinas virtuales a partir de él. Luego, utiliza la virtualización de hardware del host. En este sistema operativo Hypervisor, podemos crear y ejecutar otros sistemas operativos en él (Castillo, 2018).

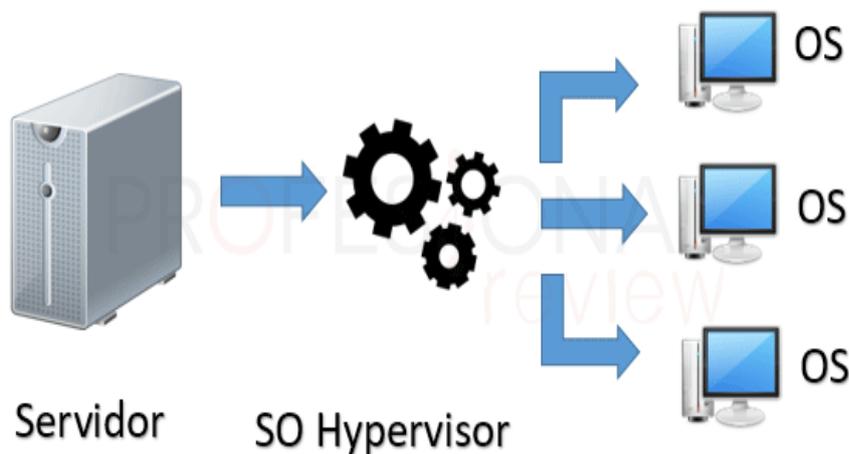


Figura 1: Infraestructura de VMware

Obtenido: <https://www.profesionalreview.com/2018/12/20/vmware-vsphere/>

2.5.3. VMware vSphere

VMware vSphere es una suite de virtualización completa diseñada para virtualizar en servidores de hardware y centros de datos. Es un entorno de virtualización nativo instalado directamente en el servidor, que se utiliza fundamentalmente para centros de

datos en empresas virtualizadas, y podemos integrarnos con VMware vCenter Server para hacer que estos centros de datos formen parte de la nube (Castillo, 2018).

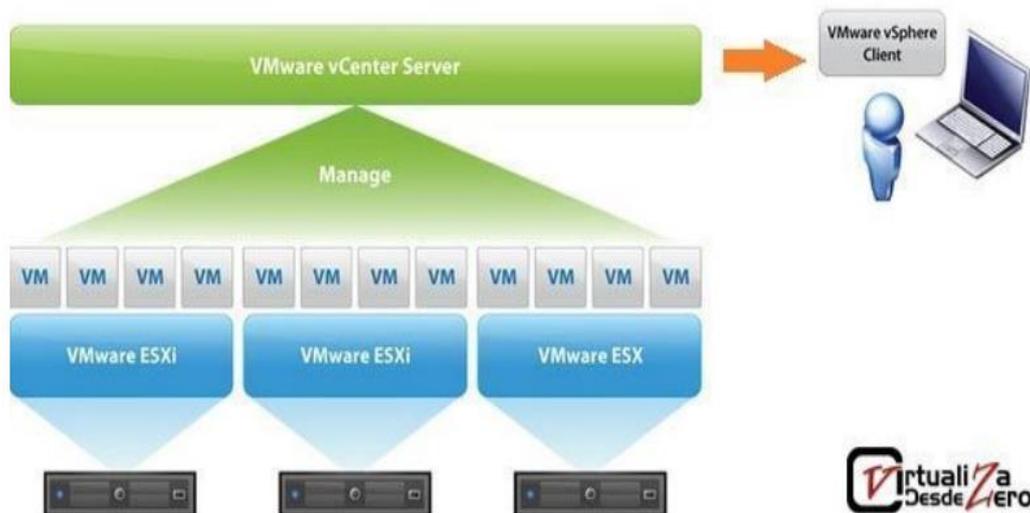


Figura 2: Infraestructura vSphere client

Obtenido: <https://virtualizadesdezero.com/que-es-vmware-vsphere/>

2.5.4. VMware vCenter Server

Esta herramienta podrá proporcionarnos una forma de administrar todas las máquinas virtuales. Con él, podremos crear, unirnos y ver clústeres de máquinas virtuales entre múltiples servidores ESXi. Por lo tanto, podemos ver todo nuestro sistema virtual en la nube, y tal vez ahora tenga una mejor comprensión del trabajo en la nube, y podamos administrarlo desde la computadora del cliente donde sea que estemos (Castillo, 2018).

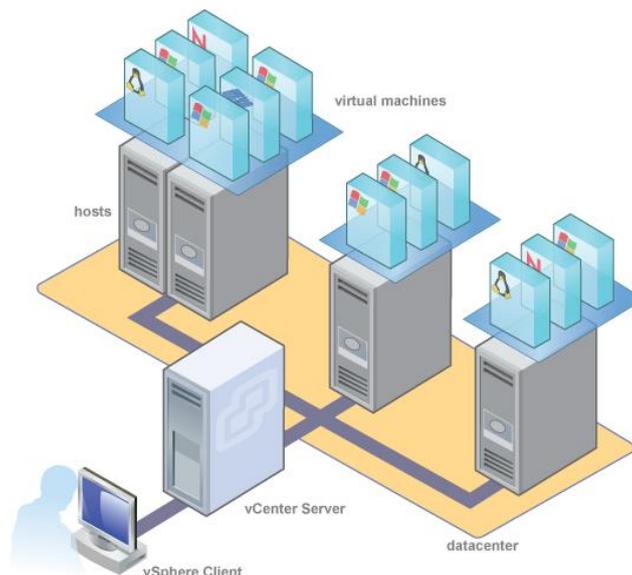


Figura 3: Infraestructura vCenter Server

Obtenido: <https://bartosha.com/the-difference-between-vsphere-esxi-vcenter-vmware/>

2.5.5. OpenFiler

Openfiler es un sistema operativo que proporciona almacenamiento adjunto a la red basado en archivos y bloqueo basado en la red del área de almacenamiento. Fue creado por Xinit Systems y se basa en una distribución de Linux. Es un software gratuito proporcionado bajo la licencia pública general GNU versión 2. Su pila es una interfaz de software con software de código abierto de terceros.

2.5.6. vSphere vMotion

VMware vSphere vMotion es una migración dinámica de un servidor a otro sin causar tiempo de inactividad de la carga de trabajo. Dependiendo de la versión de vSphere que tenga, esto se puede lograr en vSwitch, clúster o incluso en la nube. Durante la migración de la carga de trabajo, la aplicación continuará ejecutándose y los usuarios siempre tendrán acceso al sistema requerido. De esta forma se mantiene un alto nivel de productividad (vmware, 2021).

2.6. Clasificación de los Clúster

Existen varios métodos de clasificación aplicables a la clasificación de los grupos. Por ejemplo, uno de ellos considera o no considera sus propiedades propietarias, y las divide en grupos Tipo I contruidos usando componentes estándar y grupos Tipo II de fuentes propietarias (The Linux Documentation Project TLDP).

Otra clasificación ampliamente utilizada en la literatura técnica es muy útil para el análisis de conglomerados, es una clasificación para distinguir el propósito de la aplicación de la agrupación (Wikipedia Foundation WF) como lo son Alta disponibilidad y Alto rendimiento.

2.6.1. Alta disponibilidad

La redundancia permite lograr una alta disponibilidad instalando varios servidores completos (en lugar de uno solo) que pueden funcionar en paralelo y suponiendo que uno de los servidores falla, de modo que los servidores se pueden agregar y eliminar de un grupo llamado clúster. basarse en las necesidades (Linux Virtual Server LVS).

2.6.2. Alto rendimiento

Este tipo de clúster se utiliza para resolver problemas complejos que requieren mucha potencia informática. Por ejemplo, se usa para pronósticos del tiempo, simulación y encriptación (Wikipedia Foundation WF).

2.7. ADAD-SW

Describe los diferentes tipos de equilibradores de carga existentes. Más adelante, se analizará el mecanismo basado en su funcionamiento, especialmente los diferentes métodos y algoritmos de programación de distribución de carga (sin estado o con estado) ((Wikipedia Foundation WF).

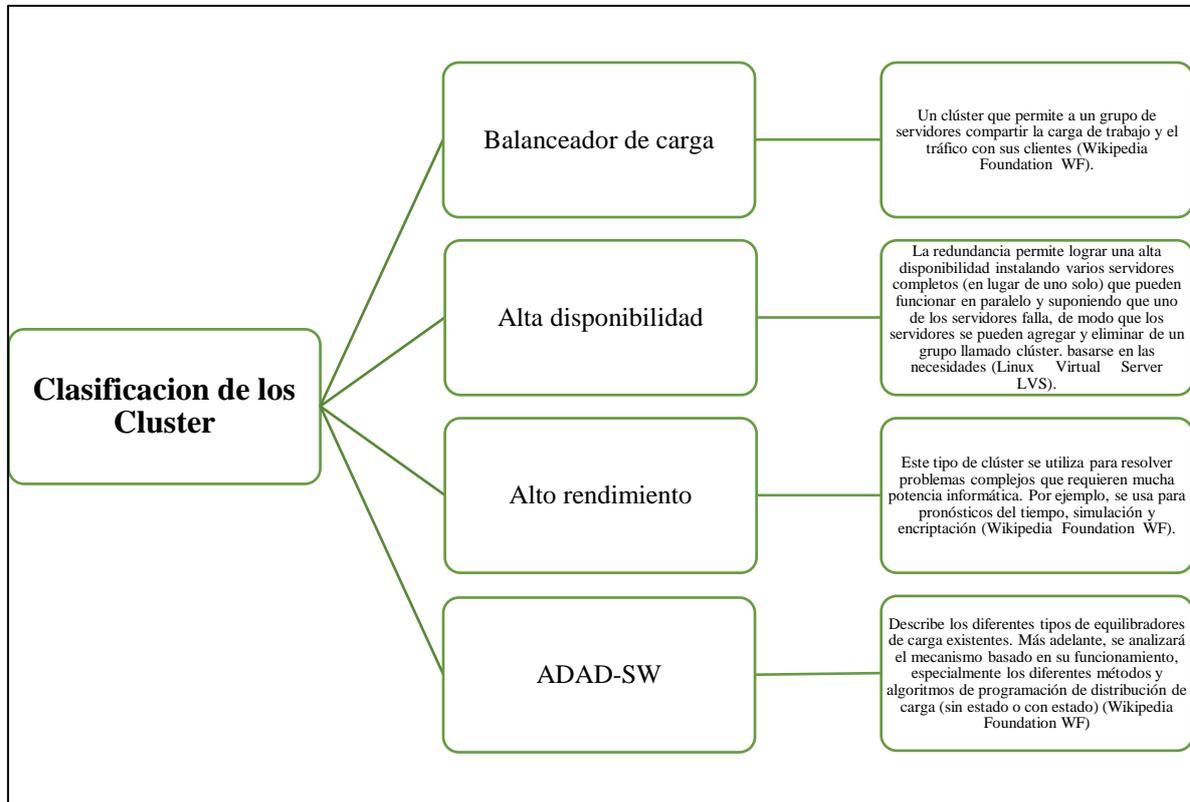


Figura 4: Clasificación de los Clúster

Elaborado por: El autor

2.8. Virtualización de Redes y Servidores Emulando Infraestructuras Tecnológicas

La virtualización es una tecnología con un gran potencial, que le permite administrar de manera efectiva el hardware, el software, la integración del servidor, el costo, el espacio físico y los recursos humanos en la infraestructura de TI, al tiempo que mejora las capacidades de administración. Y seguridad de escritorio virtual (Vilac, 2012).

Actualmente, debido a sus poderosas capacidades de almacenamiento de información y la capacidad de manejar aplicaciones y servicios a gran escala, la implementación de centros de datos en la compañía ha aumentado para optimizar los costos operativos de la infraestructura técnica. A partir del centro de datos, hoy ha surgido la llamada virtualización de servidores, Que permite instanciar múltiples máquinas virtuales en una sola máquina física (Vilac, 2012).

La zona desmilitarizada (DMZ) contiene un conjunto de patrones de diseño de red probados que pueden trabajar juntos para resolver estos problemas para los científicos. Presentamos el modelo Science DMZ, que incluye arquitectura de red, configuración de sistema, seguridad de red y herramientas de rendimiento, que crean un entorno de red científicamente optimizado (Dart, 2014).

El modelo de zona dinámica desmilitarizada (DMZ) tiene en cuenta tanto el rendimiento de la red como la seguridad. Y responde dinámicamente a las demandas de tráfico en tiempo real (Vilac, 2012)

2.9. Requerimiento de redundancia

El sistema debe implementar un mecanismo de redundancia para garantizar que repita los datos clave o el hardware que desea para evitar fallas causadas por el uso continuo (Guangping, 2009).

La tecnología de redundancia ha sido utilizada por la industria militar y aeroespacial durante muchos años para lograr una alta confiabilidad. La base de datos replicada es un ejemplo de un sistema distribuido redundante (Lopez, 2008).

Cuando los sistemas críticos deben estar disponibles y en funcionamiento las 24 horas del día, los 365 días del año, las fallas que pueden afectar el funcionamiento normal del sistema deben minimizarse. Las fallas están a punto de ocurrir, pero existen tecnologías y configuraciones que pueden ayudar a tener un sistema redundante, y algunas de ellas pueden fallar sin afectar su funcionamiento (Terrasson, Basrour, & Briand, 2009).

2.10. Requerimiento de protección y confiabilidad

El sistema debe implementar mecanismos de protección y confiabilidad para permitirle realizar estos procesos en múltiples estaciones. Estos mecanismos aseguran que, si algún sistema deja de funcionar o algunos servicios se bloquean por algún motivo, debe reemplazar inmediatamente el otro sistema y realizar las tareas del sistema anterior (Meng, 2005).

2.11. Requerimiento de implementación de clúster

Las pymes deben tener sistemas agrupados para permitirles escalar a niveles de datos, aplicaciones y servicios. Además, el sistema de clúster técnicamente admite la implementación de mecanismos de redundancia, protección, disponibilidad y confiabilidad (Song & Chokchai, 2006).

El clúster es adecuado para colecciones de computadoras o grupos empresariales creados con componentes de hardware comunes, y se comportan como una computadora. El clúster de alta disponibilidad es un clúster cuyo objetivo de diseño es proporcionar disponibilidad y confiabilidad. Estos clústeres intentan proporcionar la máxima disponibilidad de los servicios prestados (Song & Chokchai, 2006).

2.12. Requerimientos de replicación

Para las pequeñas y medianas empresas, el sistema debe admitir el mecanismo de seguridad que garantiza el funcionamiento del sistema. Cuantos más componentes, mayor es la probabilidad de problemas. Estos problemas pueden ocurrir en el servidor mismo, falla del disco, fuente de alimentación, tarjeta de red, etc. Y la infraestructura necesaria, componentes de red, acceso a Internet, sistemas, etc. del servidor que se utilizará. Luego vino el concepto de replicación, que se puede usar para replicar datos en un servidor de respaldo, proporcionando así una mayor disponibilidad en caso de inactividad esperada o inesperada del sistema. Para Py-mes, si los datos requeridos en el servidor en espera son un subconjunto de los datos requeridos en el servidor primario, el mecanismo de replicación se usa para proporcionar datos en espera semiactivos (Zhengping & Juyang, 2008).

2.13. Modelos de Virtualización

En la actualidad existen tres tipos de modelos para la virtualización que son:

El modelo de máquina virtual; el modelo paravirtual; y la virtualización a nivel de sistema operativo (McAllister, 2007).

2.14. Modelo de Máquina Virtual

El hipervisor generalmente se denomina monitor de máquina virtual (VMM), que es responsable de verificar todas las solicitudes e instrucciones del sistema virtual a la CPU y supervisar la ejecución de todos los privilegios que deben cambiarse. Dos sistemas de servidor virtual típicos son los servidores virtuales VMware y Microsoft (Kaneda, Oyama, & Yonezawa, 2005).

2.15. Modelo de Máquina Paravirtual

Al igual que las máquinas virtuales, los sistemas paravirtuales también pueden admitir diferentes sistemas operativos instalados en hardware real. Los modelos de máquina paravirtual son UML y XEN (McAllister, 2007).

2.16. Modelo de Virtualización a nivel de Sistema Operativo

Esta arquitectura elimina las llamadas del sistema entre capas y ayuda a reducir en gran medida el uso de la CPU. Además, al compartir archivos binarios y Cuando se usa una biblioteca de sistema común en la misma computadora, la posibilidad de expansión es mucho mayor, permitiendo que el mismo servidor virtual sirva a una gran cantidad de clientes simultáneamente. Los sistemas de ejemplo que utilizan la virtualización a nivel del sistema operativo son Virtuozzo y Solaris (McAllister, 2007).

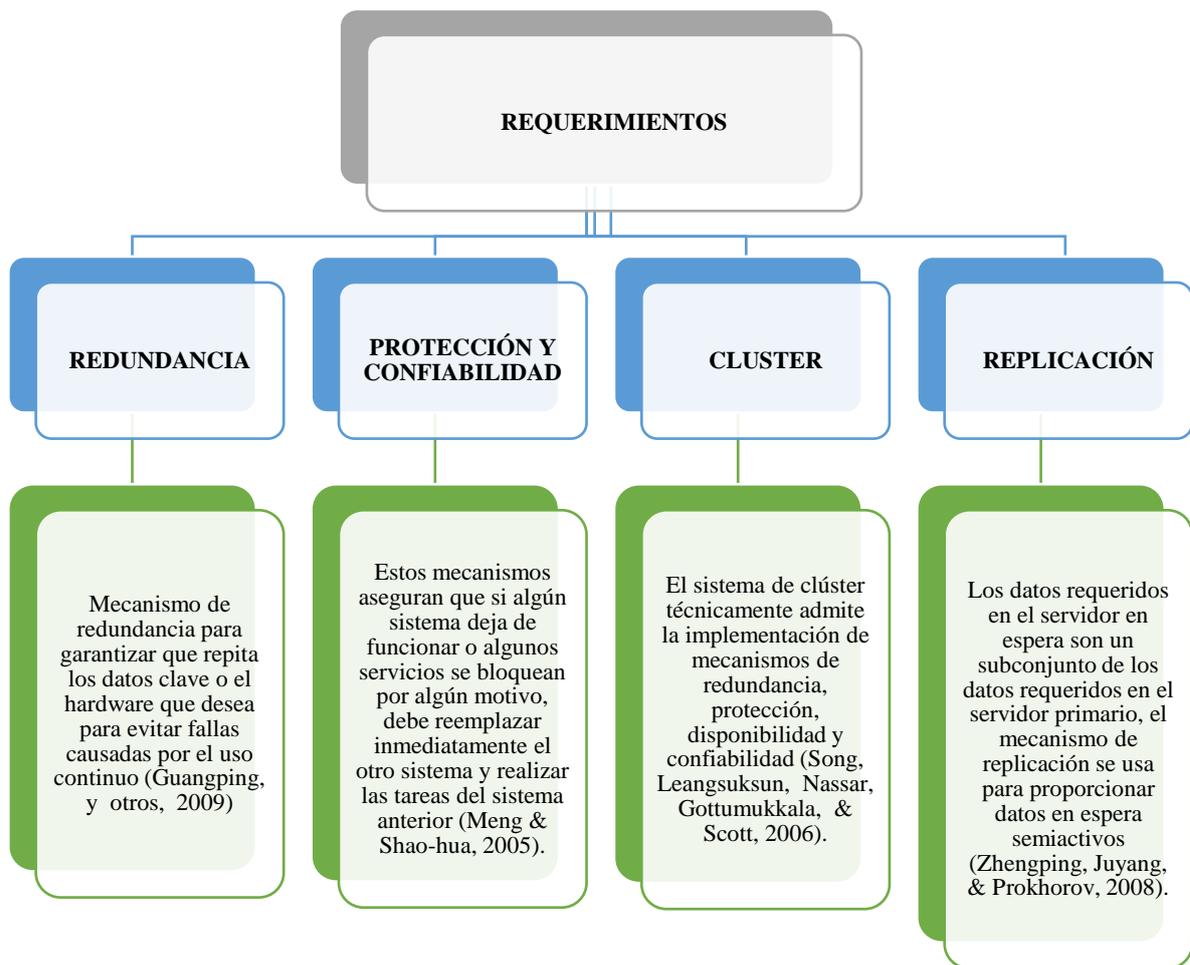


Figura 5: Requerimientos

Elaborado por: El autor

2.17. Tolerancia a Fallos para Infraestructura TI

La tolerancia a fallos es la capacidad de un sistema de permanecer operativo en caso de una falla (Avizienis, Kopetz, & Laprie, 2012). El término tolerancia a fallas fue introducido por Avizienis en 1967 y es ampliamente utilizado por la comunidad para

referirse a todo el campo de investigación que se ocupa del comportamiento de los sistemas cuando ocurren errores. Muchos términos y conceptos aún no se han consolidado y adoptado ampliamente. Diferentes grupos de investigadores usan los mismos términos de diferentes maneras y diferentes términos usan los mismos conceptos.

Varios autores se han ocupado de la nomenclatura y los conceptos básicos del rango de tolerancia a fallas. La creación de IEEE-CS TC sobre Computación tolerante a fallas en 1970 e IFIP WG 10.4 Computación confiable y tolerancia a fallas en 1980 aceleraron la aparición de un conjunto consistente de conceptos y terminología. Se presentaron siete Cartas de Intención al FTCS-12 en la Sesión Especial de 1982 sobre Conceptos Básicos de Tolerancia a Fallas, y Laprie las sintetizó en 1985 (Powell, 2001). Otro trabajo de los miembros del IFIP WG 10.4, dirigido por Lapry, se publicó en 1991 en el libro Reliability: Basic Concepts and Terminology (Laprie, Dependability: Basic Concepts and Terminology, 1992, Volume 5, 1992).

De acuerdo con Laprie (1995), según la aplicación, la garantía del rendimiento se centrará de la siguiente manera.

Tabla 1: Garantía de Rendimiento

El sistema funciona sin interrupción	Fiabilidad
El sistema no provoca fallas catastróficas	Seguridad
El sistema está disponible el mayor tiempo posible	Disponibilidad
El sistema es fácil de reparar:	Servicio (Mantenibilidad)
El sistema evita el acceso no autorizado	Confidencialidad
El sistema evita la alteración inadecuada de la información	Integridad

Elaborado por: El autor

Avizienis et al. (2004), realiza una presentación sistemática para garantizar la operatividad de los sistemas informáticos, esta se muestra en la figura 6 y consta de tres partes: problemas o daños, medidas o propiedades, y los medios por los cuales se logra una garantía de desempeño.

Los problemas o daños son circunstancias imprevistas (pero no inesperadas) que reducen la garantía de funcionamiento, es decir, hacen que los servicios prestados por el sistema no sean fiables. Se produce un error cuando el servicio proporcionado por el sistema no coincide con el especificado y, por tanto, el usuario comprende que el sistema no está funcionando correctamente. El error es una de un estado interno incorrecto del sistema. Una falla es un defecto físico o una imperfección en el hardware o software del sistema. El error es la causa del fallo (error) y esta, a su vez, es la causa del fallo. Si ocurre un error durante el tiempo de ejecución y resulta en una ejecución incorrecta de las funciones del sistema, se ha producido un fallo (Pradhan, 196) (Weber, 2003).

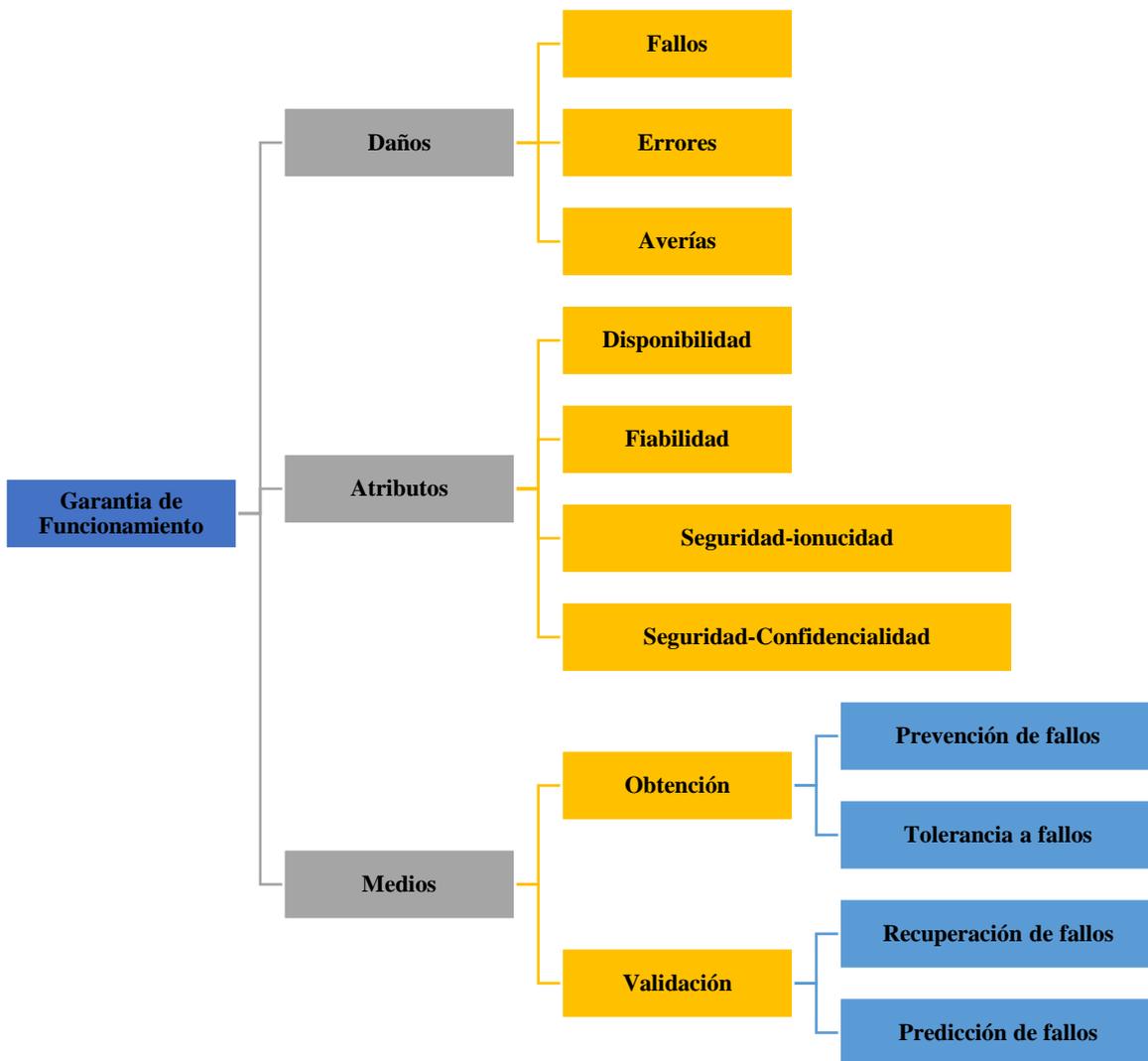


Figura 6: Garantía de funcionamiento.

Elaborado por: El autor

En la Fig. Las figuras 7, muestran una simplificación de la relación entre fallas, errores y fallas en el proceso de fabricación propuesta por Barry W. Johnson (Johnson, Inigo, & Brown, 1990). Estos pasos no se realizan al mismo tiempo, pero hay un período de inactividad, llamado retraso de error, desde el momento en que ocurre el error hasta que ocurre el error. Durante este período de tiempo de espera, la falla se considera ineficaz y la falla está latente. También se pueden especificar la latencia de detección de errores y la latencia de generación de errores.



Figura 7: Niveles de error.

Elaborado por: El autor

Dependiendo de la duración, los fallos estos se clasifican en:

- Permanentes: es un cambio irreversible en el componente.
- Temporales: presentes durante un periodo corto de tiempo. Se dividen a su vez en:
 - Transitorios: se deben a interferencias externas. La forma que aparecen y la duración son aleatorias.
 - Intermitentes: aparece de forma transitoria, pero es repetitivo, puesto que se debe a cierta combinación específica del sistema.

Los medios para obtener la garantía de funcionamiento son los métodos y técnicas. Se busca que no se produzca ninguna avería en el sistema. Para ello se aplican una serie de técnicas que conforman la tolerancia a fallos del sistema (Avizienis et al., 2004):

- Prevención de fallos (Fault avoidance): El objetivo es reducir la posibilidad de fallo del sistema, y para ello se eligen componentes de alta fiabilidad, se realiza un diseño e implementación extremadamente cuidadosa del sistema y se trata de proteger contra los agentes externos provocadores de fallos.

- Enmascaramiento de fallos (Fault masking): Una vez superada la prevención de fallos, el objetivo siguiente es que el sistema siga funcionando a pesar de la existencia de fallos. Es decir, se producen fallos, pero estos no evolucionan hacia un error. Mediante técnicas de redundancia se suministra la información necesaria al sistema para evitar los efectos de los fallos.
- Tratamiento del error: Se elimina el error antes de que produzca la avería mediante un proceso de detección, diagnóstico, aislamiento, reconfiguración y recuperación del mismo. Se construyen sistemas con redundancia dinámica (funcional o física) donde, ante la detección del error y a través de la reconfiguración, el sistema se degrada para seguir funcionando a coste de reducir su rendimiento.

Las técnicas de tolerancia a fallos están todas basadas en redundancia, exigiendo componentes adicionales o algoritmos especiales (Weber, 2003). Son implementadas generalmente por detección del error y subsiguiente recuperación del sistema (Avizienis et al., 2004). Puede haber redundancia en cualquier nivel: utilización de componentes hardware extra (redundancia en el hardware), repetición de las operaciones y comparación de los resultados (redundancia temporal), codificación de los datos (redundancia en la información) e incluso la realización de varias versiones de un mismo programa y del uso de técnicas de consistencia para comprobar que el sistema funciona correctamente (redundancia en el software). La redundancia está tan íntimamente relacionada a la tolerancia a fallos que, en la industria, la terminología usada para designar un sistema tolerante a fallos es sistema redundante.

Cuando se utiliza redundancia en el software, se debe considerar la prevención, la detección y la recuperación del sistema como un todo. La prevención consiste en generar la redundancia, la detección consiste en monitorizar y detectar el fallo y la recuperación consiste en retornar el sistema a condiciones operativas razonables, después de un fallo.

2.17.1. Ventajas

- Las técnicas de tolerancia a fallas de software brindan alta confiabilidad para componentes menos confiables.
- Simplificar la identificación y corrección de errores.
- Menores costos operativos.
- Los servidores tolerantes a fallas son más rentables de ejecutar que otras soluciones (como clústeres o matrices RAID) porque la redundancia y las

configuraciones entre los componentes están integradas en el hardware, lo que significa que las configuraciones de software son innecesarias.

- Solo se requiere una licencia de software.
- En comparación con un clúster que requiere varias licencias de aplicaciones, un servidor de conmutación por error solo requiere una licencia para cada una de estas aplicaciones. El costo de la licencia, así como el tiempo y el esfuerzo necesarios para instalar y mantener varias instancias de la misma aplicación, se reducen considerablemente (Logitek, 2021).
- Habilidades de TI menos especializadas.
- La implementación de un servidor de este tipo requiere habilidades de TI menos especializadas, por lo que ya no se requieren habilidades especializadas para implementar un sistema de alta disponibilidad.
- Menos tiempo de inactividad, menos mantenimiento.
- A medida que se reduce drásticamente el tiempo de inactividad no planificado, también se reducen los tiempos de servicio. El personal de TI puede concentrarse en otras tareas importantes (Logitek, 2021).

2.17.2. Desventajas

- Interferencia de detección de fallas en el mismo componente
- Fallo cuando se detecta un error en otro componente. Otra variación de este problema es que la tolerancia a fallos de un componente evita la detección de fallos en otro componente. Por ejemplo, si el componente B realiza algunas operaciones basadas en la producción del componente A, la tolerancia a fallas en B puede ocultar el problema en A. en el nuevo componente B. Solo después de un examen cuidadoso del sistema, queda claro que el problema realmente radica. en el componente A.
- Reducción de la prioridad de corrección de errores. Incluso si el operador es consciente del problema, un sistema tolerante a fallas puede reducir la importancia de la resolución de problemas. No corregir los errores puede provocar fallas futuras del sistema, tolerancia a fallas o fallas completas de los componentes si fallan todos los componentes redundantes.
- La complejidad de la prueba. Para algunos críticos de los sistemas tolerantes a fallas, como un reactor nuclear, no existe una manera fácil de probar el estado de los componentes redundantes. El ejemplo más famoso de esto es el desastre de

Chernóbil, cuando los operadores probaron la reserva de emergencia apagando el enfriamiento primario y secundario. La mecha falló, lo que provocó la fusión del núcleo del reactor y una liberación masiva de radiación.

- Costos. Tanto los componentes tolerantes a fallos como los componentes redundantes tienden a aumentar.
- Componentes de menor calidad. Un diseño a prueba de fallas puede permitir el uso de componentes de mala calidad, lo que puede dañar el sistema. Si bien este enfoque puede reducir el aumento de costos, el uso de algunos componentes de baja calidad puede reducir la confiabilidad del sistema a un nivel igual o peor que un sistema no tolerante a fallas.

2.17.3. Software

- **x10sure**

x10sure es una solución de aplicación y servidor Windows tolerante a fallos de Fujitsu Siemens Computers. Con este software, la aplicación se restaura a un servidor de reserva automático o de reemplazo en caso de una falla del servidor. Diseñado para empresas medianas y grandes, X10sure proporciona:

- Recuperación automatizada del sistema con requisitos mínimos de configuración, consolidación de la estructura y continuidad del negocio.
- Ofrece un riesgo de implementación bajo y una complejidad mínima para el usuario. Garantiza máxima flexibilidad, y sus usuarios logran una mayor rentabilidad a través de la implementación de servidores reales y virtuales.
- Está diseñado para integrarse en empresas con economías de escala, ya que maximiza la inversión en el futuro.

- **HPE NonStop SQL Cloud Edition**

Los sistemas HPE NonStop de la empresa Hewlett Packard Enterprise, están diseñados desde cero para entornos de misión crítica que requieren continuidad empresarial y 100% de resiliencia. NonStop elimina el riesgo de tiempo de inactividad al tiempo que satisface las necesidades de las grandes empresas al permitir el procesamiento de transacciones en línea y los requisitos de la base de datos. El entorno de software NonStop está actualmente disponible para su uso en nubes privadas.

HPE NonStop SQL Cloud Edition permite que sus aplicaciones de nube y Linux se conviertan en aplicaciones de misión crítica en minutos al abstraer la agrupación en

clústeres y la complejidad de SQL distribuido y lograr niveles más altos de disponibilidad, escalabilidad e integridad de datos.

- **NonStop OS**

NonStop OS es un sistema operativo basado en mensajes diseñado para tolerancia a fallas. Funciona con pares de procesos y garantiza que los procesos de respaldo en las CPU redundantes se hagan cargo en caso de un proceso o falla de la CPU. La integridad de los datos se mantiene durante esta transferencia, no se pierden ni dañan transacciones o datos.

El sistema operativo completo se llama NonStop OS e incluye la capa Guardian, que es el componente de bajo nivel del sistema operativo, y la llamada personalidad OSS, que se ejecuta en esta capa, que implementa una interfaz tipo Unix para usar otra. sistema. los componentes están listos para funcionar.

El sistema operativo y las aplicaciones están diseñados para admitir hardware tolerante a fallas. El sistema operativo supervisa continuamente el estado de todos los componentes y modifica los controles según sea necesario para mantener el rendimiento. El software también tiene funciones integradas que le permiten escribir programas como programas fácilmente disponibles.

Esto se logra mediante el uso de múltiples procesos, uno de los cuales realiza todo el procesamiento principal y el otro actúa como una "copia de seguridad en caliente" que recibe actualizaciones cuando el principal alcanza un punto crítico de procesamiento. Cuando se detiene la conexión principal, la copia de seguridad comienza a funcionar con la transacción actual.

- **Stratus everRun**

El programa informático everRun Enterprise es un software de prevención del tiempo de inactividad que permite la continuidad del negocio y la integridad de los datos manteniendo las aplicaciones en funcionamiento con disponibilidad de tipo mainframe en servidores x86.

La aplicación se encuentra en dos máquinas virtuales. Si una máquina falla, la aplicación continúa ejecutándose en la otra máquina con sin interrupciones ni pérdida de datos. Si un componente falla, se reemplaza con el componente saludable del segundo sistema. La E / S se refleja automáticamente en el servidor redundante. La comprobación de memoria asegura todas las transacciones en vuelo como, así como los datos en la memoria y la caché se conservan, si la máquina se reinicia.

CAPÍTULO III

METODOLOGÍA

El presente estudio es de tipo cuasi-experimental orientado a la investigación de la técnica de tolerancia a fallos, usada en data center virtuales, para lo cual será implementada en un ambiente de prototipo de escenarios de prueba usando tecnología VMware.

3.1 Tipo y Diseño de Investigación

Los tipos de estudio empleados en el presente proyecto de investigación son:

- **Método Analítico:** El método de estudio se fundamenta en el análisis de la técnica de tolerancia a fallos en los servidores alojados en el data center virtualizado de nuestro escenario y sus componentes.
- **Según la intervención del investigador,** este estudio es cuasi-experimental, porque se va a realizar una simulación de cómo funciona la técnica de tolerancia a fallos, para lograr una alta disponibilidad de servicios, mediante un escenario de pruebas.
- **Según la planificación de la medición de la variable,** el tipo de estudio es retrospectivo porque los datos se escogen según el sistema.
- **De acuerdo con el número de mediciones de la variable de estudio,** es transversal porque los datos se van a medir una sola vez.
- **Según al número de variables de interés,** es descriptivo porque se va a analizar la técnica de tolerancia a fallos en un data center virtualizado.

3.2 Población y Muestra

Esta investigación no cuenta con población y muestra ya que ya que se desarrollará en un entorno de escenario de pruebas prototipo utilizando tecnología VMware.

3.3. Operacionalización de las variables

Tabla 2: Matriz de consistencia

PROBLEMA	TEMA	OBJETIVOS	INDICADORES
<p>Alta disponibilidad de los recursos existentes en los servidores que forman parte de un data center virtualizado con la implementación de la técnica de tolerancia a fallos usando tecnología VMware</p>	<p>Implementación de un Data Center virtual para la aplicación de la técnica de tolerancia a fallos en servidores usando tecnología VMware.</p>	<p>Implementar la técnica de tolerancia a fallos para servidores que forman parte del clúster de un data center apoyado en la tecnología VMware.</p> <p>O. Específicos</p> <ol style="list-style-type: none"> 1. Realizar un estudio sobre la técnica de Tolerancia a Fallos para infraestructuras TI. 2. Implementar mediante virtualización una infraestructura tecnológica TI o Data Center con clúster de servidores utilizando VMware para la configuración de la técnica de Tolerancia a fallos. 3. Evaluar la aplicación de Tolerancia a fallos en un escenario de pruebas. 	<p>-Tiempo de conformación del servidor (espejo) de respaldo.</p> <p>-Tiempo de levantamiento del servidor replica.</p> <p>-Disponibilidad de los recursos existentes en los servidores que forman parte de un data center virtualizado.</p>

Elaborado por: El autor

3.4 Procedimientos

Para la Implementación de la técnica de tolerancia a fallos para servidores que forman parte de un clúster en un data center virtualizado usando tecnología VMware, el proceso a seguir es el siguiente:

Paso 1: Recopilación de información

Paso 2: Instalación y configuración de máquinas virtuales

Paso 3: Implementación de la técnica tolerancia a fallos en un escenario de pruebas de VMware, para los cual se realiza el desarrollo siguiente:

- Verificación de la Máquina Host (herramientas de verificación)
- Habilitación de VTX HARDWARE para virtualización
- Instalación VMware Workstation 15
- Instalación Servidores ESX/ESXi 5.1
- Instalación Servidor vCenter Server Appliance 5.1
- Instalación vSphere 5.1
- Instalación y Configuración de una Cabina de discos (OPENFILER)
- Creación del Centro de Datos e Integración de servidores ESX/ESXi
- Agregar a los servidores ESXi, S.O Windows y Linux
- Configuración de un Almacenamiento Común (VKERNEL) en VM de ESX/ESXi
- Creación de un CLÚSTER y Configuración de Alta Disponibilidad (HA) y Fault Tolerance
- Análisis de los resultados.

3.5 Procedimientos y análisis

La implementación de la técnica de tolerancia a fallos aplicada en un escenario de pruebas virtualizado mediante tecnología VMware nos permitirá comprobar la alta disponibilidad que esta nos otorga y la rápida respuesta de los servidores ante cualquier fallo.

CAPÍTULO IV

La implementación del datacenter virtual, clúster, servidores, Vmotion y técnica de tolerancia a fallos se encuentra detallada en el manual de implementación realizada por el autor adjunto a este trabajo.

RESULTADOS Y DISCUSIÓN

4.1. Migración de una máquina virtual a través de vMotion

La figura 8, muestra la opción de migración de una máquina virtual de un servidor hacia otro con la maquina encendida.



Figura 8: Migración de máquinas virtuales

4.2. Migración entre servidores

La figura 9, muestra las opciones que tenemos para migrar la máquina virtual, en este caso podemos migrar del servidor 192.168.145.128 al servidor 192.168.145.40.

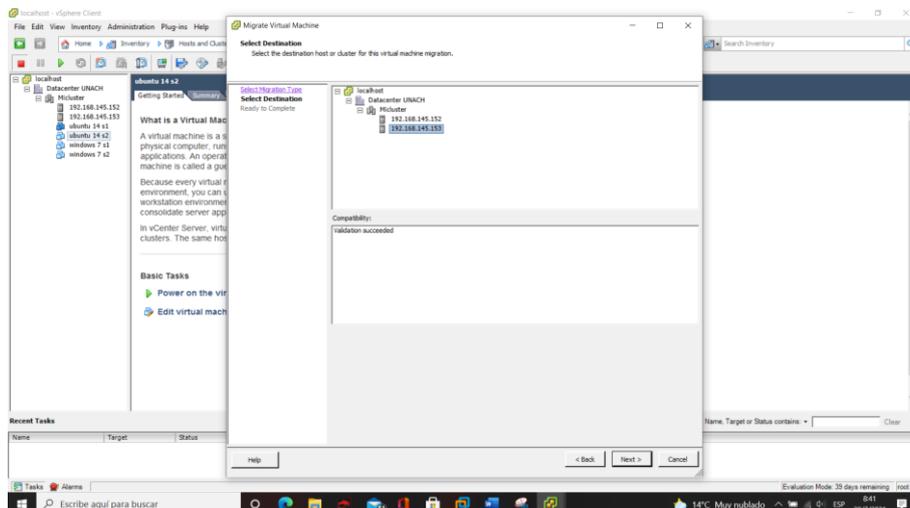


Figura 9: Selección de servidor

4.3. Aplicación de la técnica tolerancia a fallos

La figura 10, indica la conformación del servidor espejo en el segundo servidor debido a la implementación de la técnica de tolerancia a fallos.

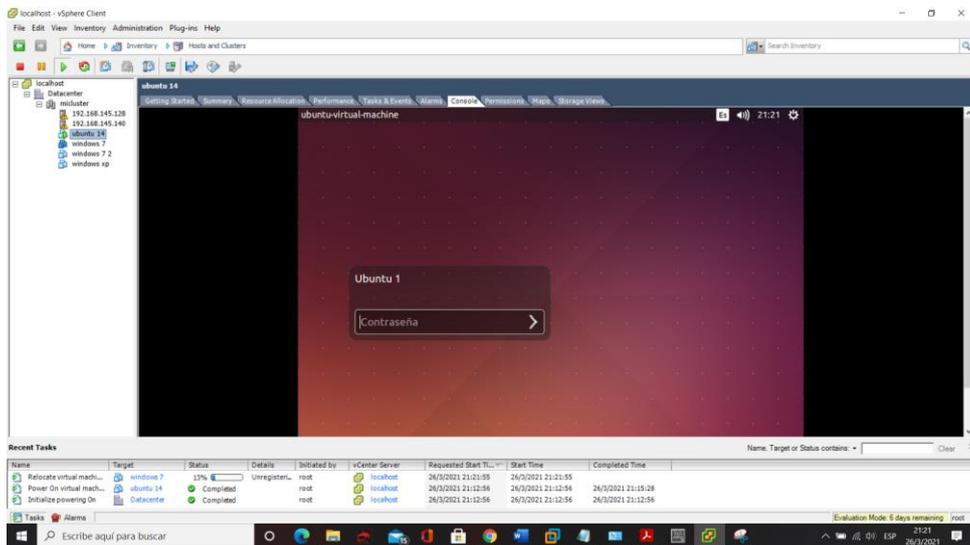


Figura 10: Conformación del servidor espejo.

4.4. Tiempo de conformación del servidor espejo.

La figura 11, indica el tiempo que tarda la conformación del servidor espejo en el segundo servidor, en este caso empezó a las 19:27:52 y termino a las 19:55:24 con un total de 27 minutos con 32 segundos.

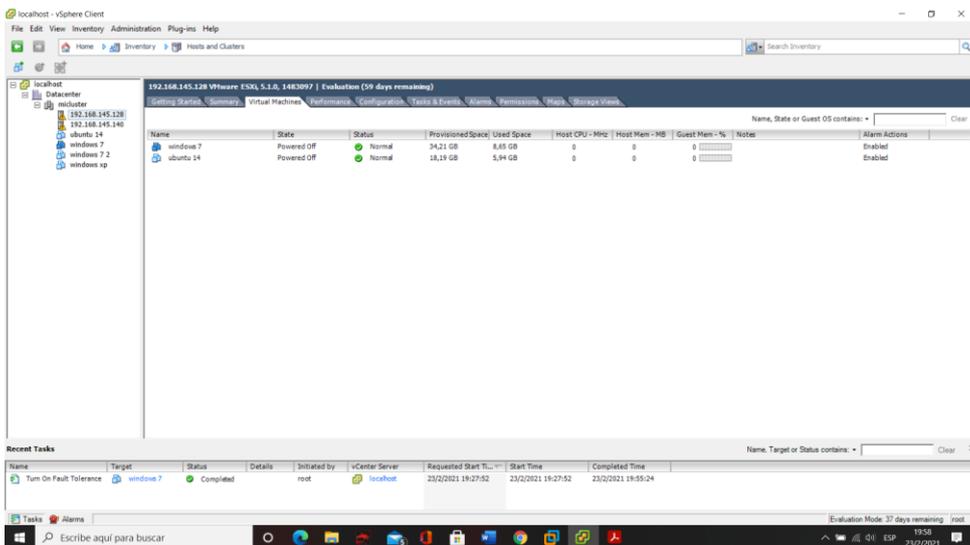
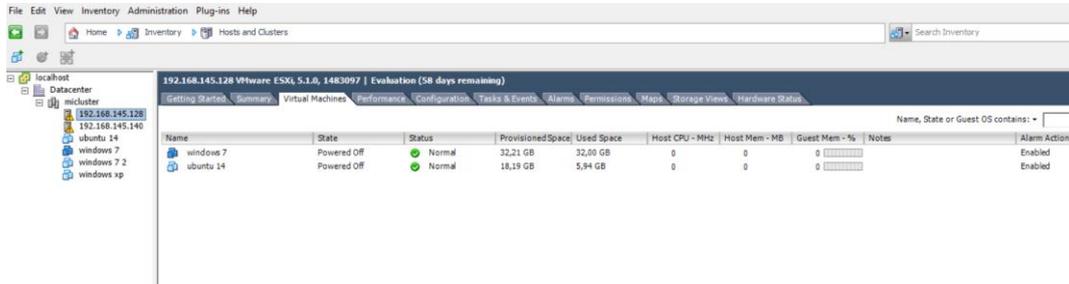


Figura 11: Tiempo de conformación del servidor espejo.

4.5. Técnica Tolerancia a fallos aplicada en la maquina Windows 7

En la figura 12, se muestra la máquina virtual Windows 7 de color azul debido a que se le aplico la técnica de tolerancia a fallos.

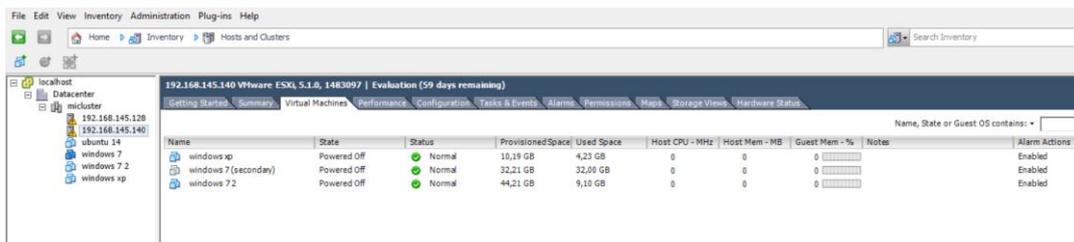


Name	State	Status	Provisioned Space	Used Space	Host CPU - MHz	Host Mem - MB	Guest Mem - %	Notes	Alarm Actions
windows 7	Powered Off	Normal	32,21 GB	32,00 GB	0	0	0		Enabled
ubuntu 14	Powered Off	Normal	18,19 GB	5,94 GB	0	0	0		Enabled

Figura 12: Maquina Windows 10 aplicada técnica tolerancia a fallos.

4.6. Verificación del servidor espejo en el segundo servidor

La figura 13, muestra el espejo creado de la maquina Windows 7 en el segundo servidor.

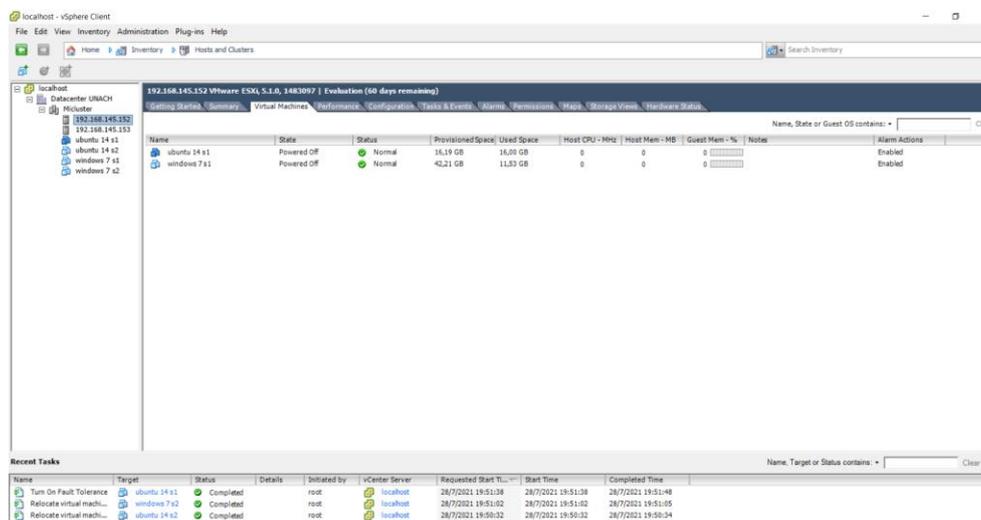


Name	State	Status	Provisioned Space	Used Space	Host CPU - MHz	Host Mem - MB	Guest Mem - %	Notes	Alarm Actions
windows xp	Powered Off	Normal	10,19 GB	4,23 GB	0	0	0		Enabled
windows 7 (secondary)	Powered Off	Normal	32,21 GB	32,00 GB	0	0	0		Enabled
windows 7.2	Powered Off	Normal	44,21 GB	9,10 GB	0	0	0		Enabled

Figura 13: Espejo creado en el segundo servidor.

4.7. Funcionamiento de la técnica de tolerancia a fallos aplicada en la maquina Ubuntu 14 S1

La figura 14, muestra las máquinas virtuales que se encuentran alojadas en el primer servidor.



Name	State	Status	Provisioned Space	Used Space	Host CPU - MHz	Host Mem - MB	Guest Mem - %	Notes	Alarm Actions
ubuntu 14 s1	Powered Off	Normal	18,19 GB	16,08 GB	0	0	0		Enabled
windows 7s1	Powered Off	Normal	42,21 GB	11,83 GB	0	0	0		Enabled

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Time	Start Time	Completed Time
Turn On Fault Tolerance	ubuntu 14 s1	Completed		root	localhost	28/7/2021 19:51:38	28/7/2021 19:51:38	28/7/2021 19:51:48
Relocate virtual mach...	windows 7s2	Completed		root	localhost	28/7/2021 19:50:02	28/7/2021 19:50:02	28/7/2021 19:50:05
Relocate virtual mach...	ubuntu 14 s2	Completed		root	localhost	28/7/2021 19:50:32	28/7/2021 19:50:32	28/7/2021 19:50:34

Figura 14: Máquinas virtuales primer servidor.

La figura 15, muestra las máquinas virtuales que se encuentran alojadas en el segundo servidor además del espejo de Ubuntu 14 que es la máquina que tiene activada la técnica de tolerancia a fallos.

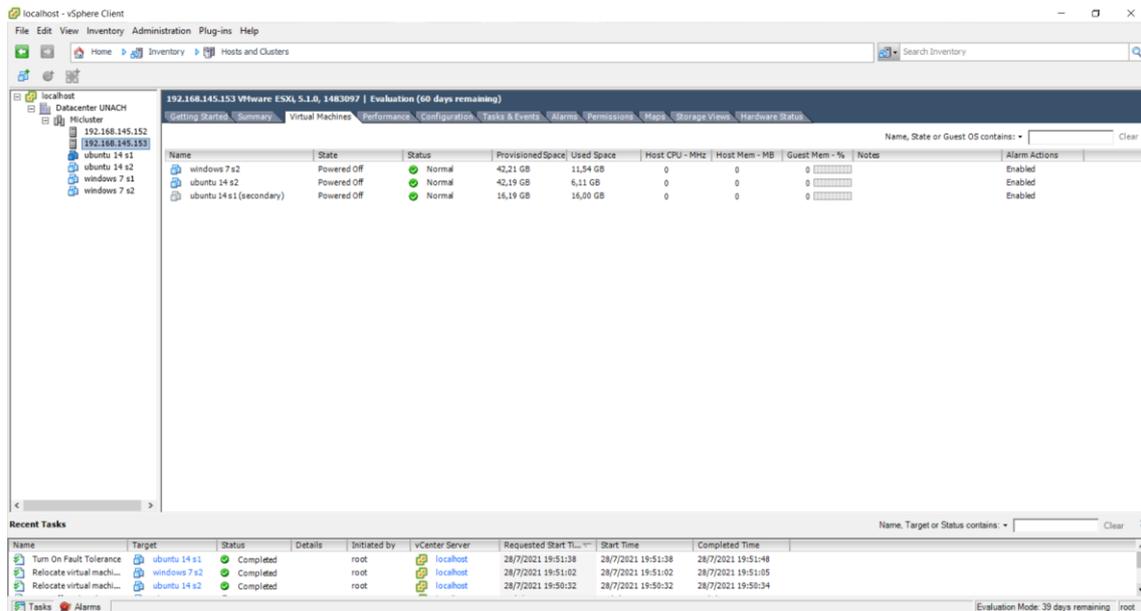


Figura 15: Máquinas virtuales segundo servidor.

La figura 16, indica la entrada en modo mantenimiento del primer servidor.

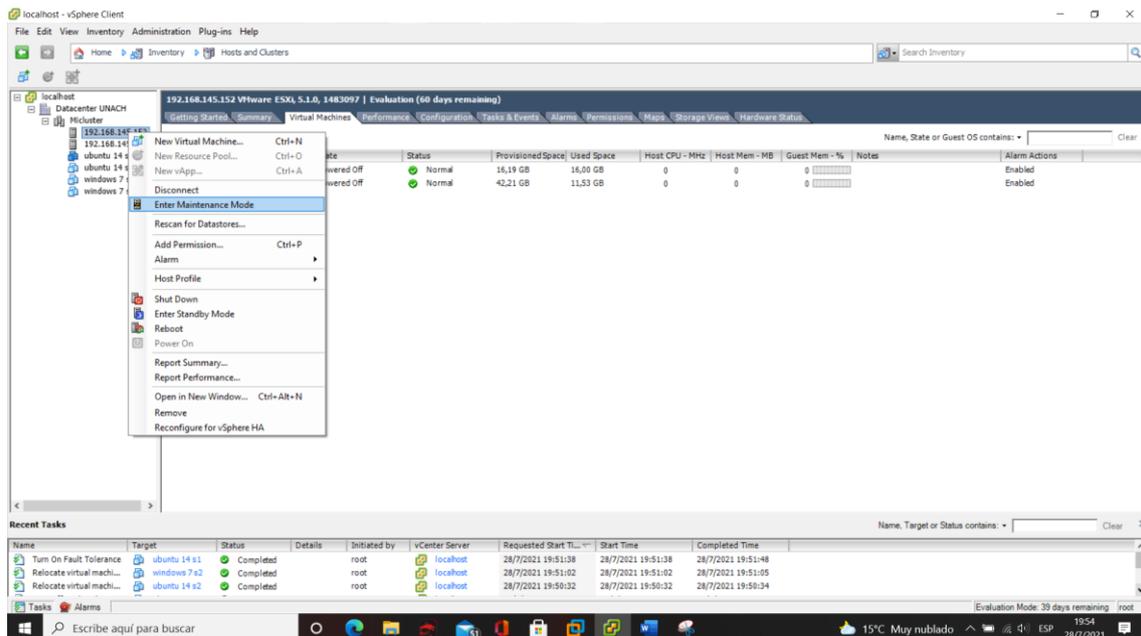


Figura 16: Modo mantenimiento servidor 1.

La figura 17, muestra la migración de las máquinas virtuales del servidor uno al servidor 2 debido a que el servidor entro en modo mantenimiento.

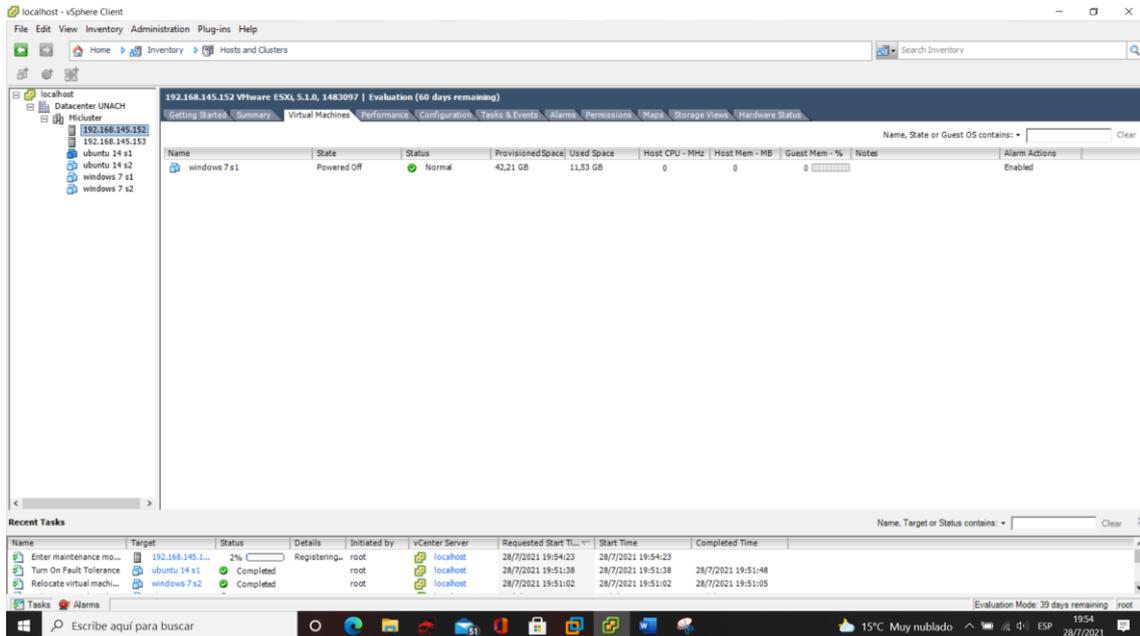


Figura 17: Migración máquinas virtuales servidor 1.

La figura 18, indica todas las máquinas virtuales que se encuentran en el servidor 2 después de que el servidor 1 entro en modo de mantenimiento.

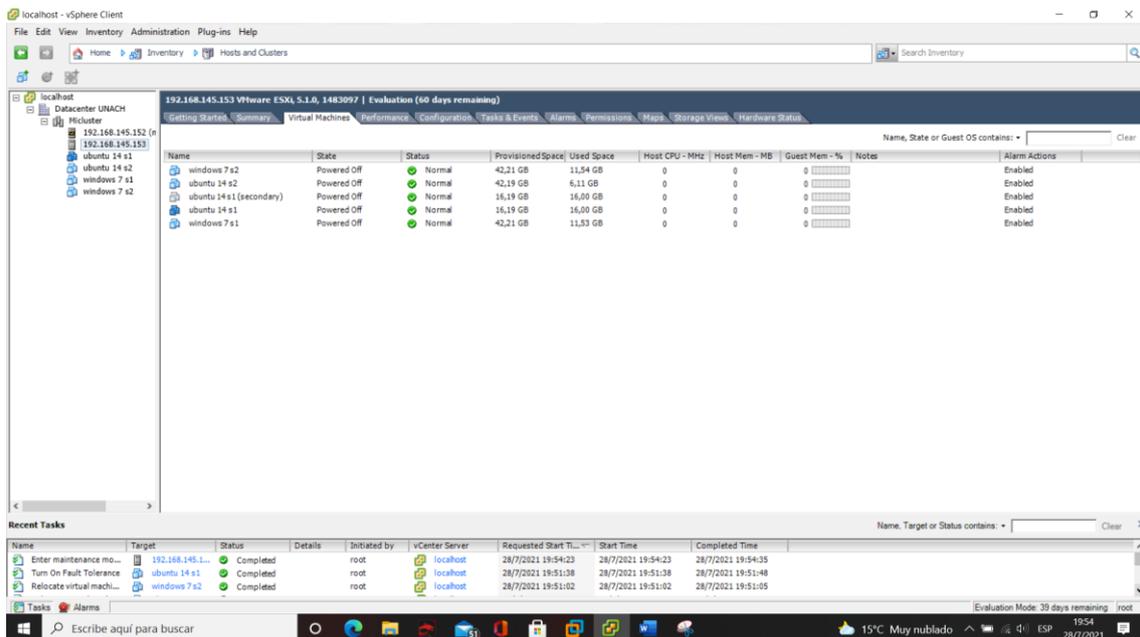


Figura 18: Máquinas virtuales alojadas en el servidor 2 después de modo mantenimiento servidor 1.

La figura 19, indica la entrada en modo mantenimiento del segundo servidor.

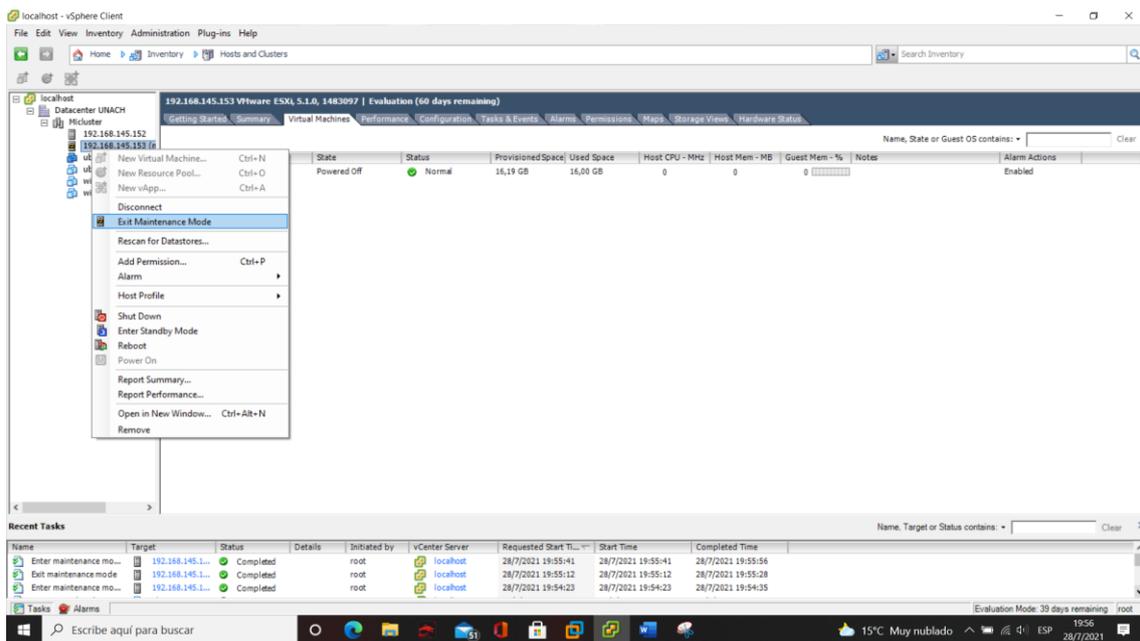


Figura 19: Modo mantenimiento servidor 2.

La figura 20, indica las máquinas virtuales que se encuentran en el segundo servidor después de que este entrara en modo mantenimiento, como se puede observar solo quedo la máquina que tiene activa la técnica de tolerancia a fallos.

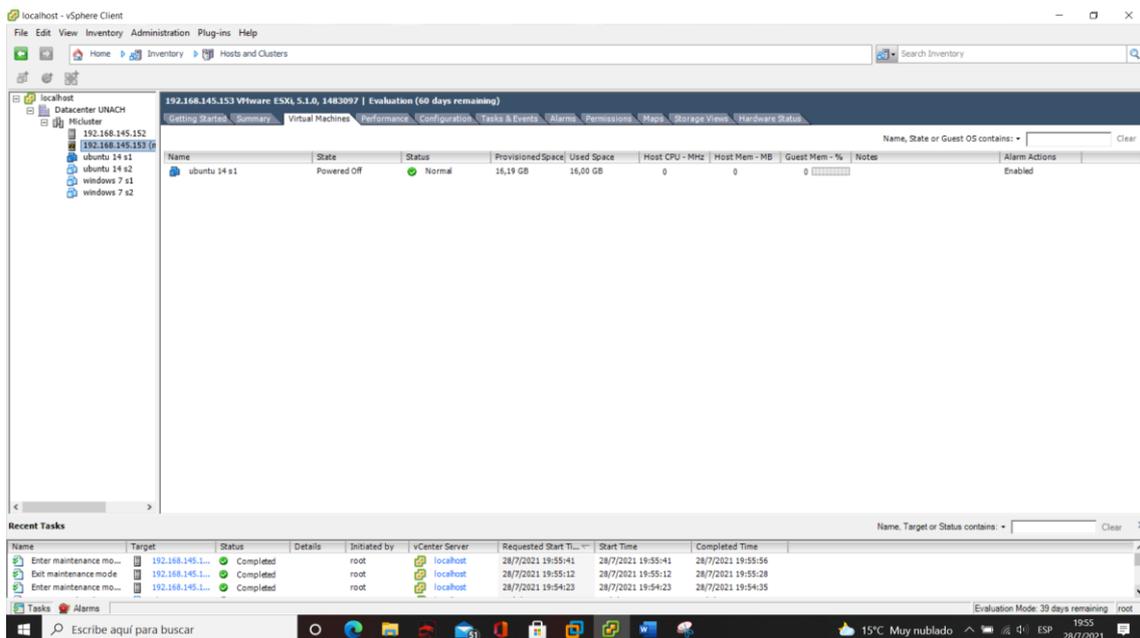


Figura 20: Máquina activada la tolerancia a fallos en servidor 2.

La figura 21, muestra las máquinas virtuales que se alojaron en el primer servidor después de que el primero entro en modo mantenimiento, entre las maquinas se encuentra el espejo de la maquina Ubuntu 1 s1 que es la que tiene activa la técnica de tolerancia a fallos.

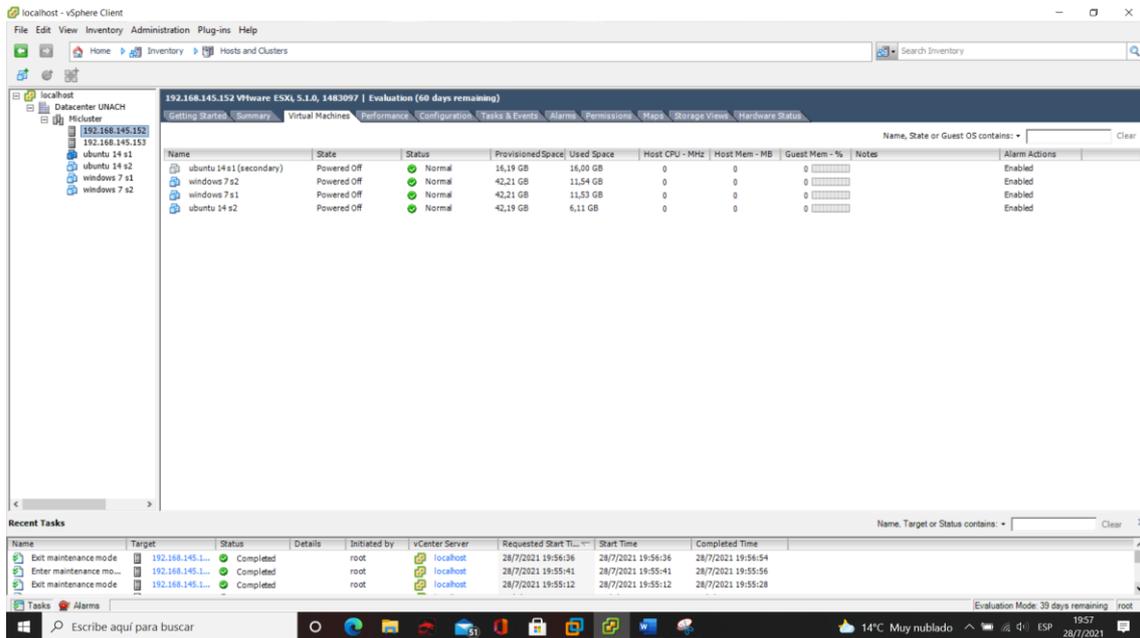


Figura 21: Espejo de Ubuntu 14 s1 en servidor 2.

CONCLUSIONES

- Al crear un escenario virtualizado se pudo disminuir el costo de implementación de un data center, convirtiéndose en una de las mejores opciones para las pequeñas y medianas empresas.
- Con vMotion se logró una transferencia de máquina virtual en caliente de un servidor a otro, logrando así una alta disponibilidad de los servicios.
- Al implementar la técnica de tolerancia a fallos (fault tolerance) se creó un espejo en el segundo servidor el cual se va a activar ante cualquier fallo del principal logrando así una alta disponibilidad.
- Las herramientas utilizadas facilitaron la implementación de datacenter virtuales menorando el costo y mejorando la funcionalidad.

RECOMENDACIONES

- Se recomienda verificar que estén habilitadas las opciones de virtualización de la computadora que va a ser utilizada para la creación de la infraestructura virtual.
- Para implementar toda la infraestructura virtual, se recomienda utilizar las siguientes herramientas por el tema de compatibilidad:
 - Mínimo de 16 de RAM
 - Workstation 15
 - Servidores ESXi 5.1
 - vCenterServer 5.1
 - vSphere Client 5.1
 - Openfiler.
 - Sistemas Operativos (Windows- Ubuntu).
- Se recomienda la adquisición de licencias de las herramientas a utilizar ya que la mayoría solo tiene un periodo de prueba gratis.
- Se recomienda que una vez creada la máquina virtual se retire la ISO ya que esto puede generar problemas en las configuraciones siguientes.
- Se recomienda tener la computadora conectada a un UPS ya que en caso de apagones este evitara que es computador se apague y se pierda tanto información como los servicios.

BIBLIOGRAFÍA

- Avizienis, A., Kopetz, H., & Laprie, J.-C. (2012). *The Evolution of Fault-Tolerant Computing: In the Honor of William C. Carter*. Baden: Springer-Verlag Wien New York.
- Avizienis, A., Laprie, J., Randell, B., & Landwehr, C. (2004). *Basic concepts and taxonomy of dependable and secure computing*. Los Angeles.
- Castillo, J. (20 de Diciembre de 2018). *www.profesionalreview.com*. Obtenido de <https://www.profesionalreview.com/2018/12/20/vmware-vsphere/>
- Chuquiguanca, L. (05 de enero de 2015). *dspace*. Obtenido de <http://dspace.unl.edu.ec:9001/jspui/bitstream/123456789/11025/3/Chuquiguanca%20Vicente%2C%20Leonardo%20Rafael%2C%20Malla%20Bustamante%2C%20Edyson%20Javier.pdf>
- Dart, E. (14 de agosto de 2014). *iee*. Obtenido de <https://ieeexplore.ieee.org/document/6877518>
- Galán, A. (12 de febrero de 2015). *Revista Ingeniería y Ciencia de la Facultad de Ingeniería de la Universidad Rafael Landívar*. Obtenido de <http://168.234.75.179/index.php/riyc/article/view/1028/911>
- Guangping, X. (15 de enero de 2009). *researchgate*. Obtenido de https://www.researchgate.net/publication/220909108_Model_and_Evaluation_of_Redundant_Data_Organization_for_High_Availability_in_Structured_Overlay_Networks
- Guirao, A. (25 de enero de 2011). *upcommons*. Obtenido de <https://upcommons.upc.edu/handle/2099.1/11289>
- Johnson, W., Inigo, R., & Brown, K. (1990). *Design, implementation, and testing of an adaptable optimal controller for an electric wheelchair*. Los Angeles: IEEE Transactions on Industry Applications .
- Kaneda, K., Oyama, Y., & Yonezawa, A. (16 de enero de 2005). *researchgate*. Obtenido de

https://www.researchgate.net/publication/228573206_Virtual_machine_monitor_for_providing_a_single_system_image

Laprie, J. (1992). *Dependability: Basic Concepts and Terminology, 1992, Volume 5*. Viena: Springer-Verlag Viena.

Laprie, J. (1995). *Dependable Computing and Fault Tolerance: Concepts and Terminology*. Viena: FTCS-1.5.

Lemus, I. (10 de marzo de 2020). *conocimientolibre*. Obtenido de <https://conocimientolibre.mx/historia-de-la-virtualizacion/>

Logitek. (2021). *www.altadisponibilidadlogitek.com*. Obtenido de Tolerancia a Fallos: <https://www.altadisponibilidadlogitek.com/tecnologias-high-availability/tolerancia-a-fallos/#:~:text=Los%20servidores%20tolerantes%20a%20fallos,menor%20coste%20que%20otras%20opciones>.

Lopez, E. (12 de diciembre de 2008). *researchgate*. Obtenido de https://www.researchgate.net/publication/224354387_A_High_Availability_Firewall_Model_Based_on_SCTP_Protocol

Martín, D., Marrero, M., Urbano, J., Barra, E., & José-Antonio. (10 de junio de 2011). *e-archivo*. Obtenido de https://e-archivo.uc3m.es/bitstream/handle/10016/21643/virtualizacion_EPI_2011.pdf?sequence=2&isAllowed=y

McAllister, N. (12 de febrero de 2007). *infoworld*. Obtenido de <https://www.infoworld.com/article/2661737/server-virtualization-under-the-hood.html>

Meng, J. (16 de octubre de 2005). *researchgate*. Obtenido de https://www.researchgate.net/publication/4195520_Modeling_and_analysis_of_high-availability_routing_software

Pessolani et al. (18 de mayo de 2012). <http://sedici.unlp.edu.ar/>. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/18375/Documento_completo.pdf?sequence=1&isAllowed=y

- Powell, D. (2001). *A Generic Fault-Tolerant Architecture for Real-Time Dependable Systems*. Paris: Springer Science.
- Pradhan, D. (196). *Fault Tolerant System Design*. New Jersey: Prentice Hall.
- RedHat. (19 de junio de 2018). *www.redhat.com*. Obtenido de <https://www.redhat.com/es/topics/virtualization/what-is-virtualization>
- Rodriguez, G. (16 de octubre de 2013). *driverlandia*. Obtenido de <https://www.driverlandia.com/crear-servidor-de-virtualizacion-con-vmware-esxi/>
- Song, H., & Chokchai, L. (19 de noviembre de 2006). *researchgate*. Obtenido de https://www.researchgate.net/publication/255585019_Availability_Modeling_and_Evaluation_on_High_Performance_Cluster_Computing_Systems
- Terrasson, G., Basrour, S., & Briand, R. (26 de septiembre de 2009). *researchgate*. Obtenido de https://www.researchgate.net/publication/221187290_A_Top-Down_Approach_for_the_Design_of_Low-Power_Microsensor_Nodes_for_Wireless_Sensor_Network
- Vilac. (19 de abril de 2012). *repositorio.espe*. Obtenido de <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/5161/T-ESPE-033141-A.pdf?sequence=2&isAllowed=y>
- vmware. (2019 de julio de 2019). *www.vmware.com*. Obtenido de <https://www.vmware.com/latam/timeline.html>
- vmware. (19 de enero de 2021). Obtenido de <https://www.vmware.com/latam/products/vsphere/vmotion.html>
- VMWARE. (9 de ENERO de 2021). *www.vmware.com*. Obtenido de <https://www.vmware.com/co/products/workstation-pro/faq.html>
- Weber, T. (2003). *Tolerância a falhas: conceitos e exemplos*. Brasilia: Instituto de Informática - UFRGS.
- Zhengping , J., & Juyang , W. (10 de julio de 2008). *cse*. Obtenido de http://www.cse.msu.edu/~weng/research/ICDL08_0077.pdf

ANEXOS

Se adjunta el manual de configuración.