

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**VIRTUALIZACIÓN DE SERVIDORES MEDIANTE SOLARIS CONTAINERS**

**TESIS**

QUE PARA OBTENER EL TÍTULO DE:

**INGENIERO EN COMPUTACIÓN**

PRESENTAN :

**CUEVAS RESÉNDIZ BETTINA MARGARITA**

**MARTÍNEZ RAMÍREZ RICARDO ERNESTO**

**MIER GARCÍA IRASED URFILA**

**RODRÍGUEZ SAAVEDRA RITA ELIZABETH**

DIRECTORA:

**ING. GLORIA MATA HERNÁNDEZ**



**CIUDAD UNIVERSITARIA, 2011**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Esta hoja fue dejada en blanco a propósito.

## **AGRADECIMIENTOS**

Antes que nada, doy gracias a la UNAM y a la Facultad de Ingeniería por brindarme los recursos y formarme profesionalmente, por las experiencias ahí vividas tanto educativas, como deportivas y culturales. Pero sobre todo, gracias porque de no ser por la UNAM no habría llegado tan lejos.

Gracias a la Ing. Gloria Mata por guiarnos en la realización de esta Tesis y por ayudarnos a concluirla.

Gracias a mi familia por su amor y cariño, a mis padres por darme siempre todo lo que pudieron, por su comprensión, compañía y enseñanzas. A mi Mamá por trabajar siempre tanto, por enseñarme cómo salir adelante y por todo el amor que me da cada día.

A mis amigos y amigas: Elisa, Irased, Lilia, Denisse, Gaby, Rafa, Nancy, Martín y demás por formar parte de mi vida, por todas las experiencias que compartimos juntos y por su apoyo. A Martín, muchas gracias por ayudarnos a la realización de este trabajo de Tesis, sin tu apoyo no habríamos podido lograrlo. A Nancy, gracias por cuidar nuestro trabajo.

A mis compañeros de Tesis por su trabajo, dedicación y cooperación.

Gracias Rita por ser mi compañera en clases y en este trabajo de Tesis, pero sobre todo por ser mi mejor amiga, por tu apoyo, comprensión y cariño, porque me ubicas y me cuidas, por siempre estar ahí conmigo cuando te necesito y gracias por todos los momentos y experiencias que has compartido conmigo.

Bettina M. Cuevas Reséndiz

"La culminación de este trabajo de tesis no solo es producto del esfuerzo de los cuatro integrantes del equipo, sino de muchas personas que estuvieron alrededor. En especial quiero agradecer a mi familia por su apoyo incondicional y muchas veces alentador a seguir adelante. Así mismo a todas aquellas personas que me han acompañado a lo largo de la vida. - Y aquí me veo, pobre loco, sin saber más que al principio - (Fausto)"

Gracias!!

Ricardo Ernesto Martinez Ramirez

**A mis Padres**

Gracias por su cariño, comprensión y apoyo sin condiciones ni medida a lo largo de este logro profesional que finamente culmina. Gracias por guiarme sobre el camino de la educación.

**A mis Hermanos**

Por sus comentarios, sugerencias y opiniones.

**A la Universidad**

Por haberme dado la oportunidad de estudiar esta carrera, la cual me ha aportado grandes beneficios tanto en mi vida personal como profesional.

**A mis Compañeros de Tesis**

Por su compañerismo y Motivación.

Irased Mier García

Quiero agradecer, antes que nada, a mi familia. Agradezco a mis padres todo el apoyo y amor que me han brindado desde el momento en que nací, gracias por todo el tiempo que invirtieron en mí y por todos sus cuidados, por sus enseñanzas; gracias por esforzarse siempre en darme lo mejor, gracias por guiarme, gracias por toda la cultura y moral que cultivaron en mí, gracias por sus ideales y muchas gracias por esmerarse en hacerme una buena persona y ser humano. A mi hermana también le agradezco todo su apoyo y toda la tolerancia que ha tenido conmigo, gracias por sus buenos consejos, gracias por cuidarme, gracias por todos los momentos juntas y gracias por ayudarme a dedicarle todo el tiempo posible a la Tesis, te quiero mucho. Gracias a Teddy y a Maggie por levantarme el ánimo siempre y recibirme diario con gran entusiasmo.

Agradezco a la Universidad Nacional Autónoma de México, por brindarme el gran orgullo de formar parte de su alumnado, por enseñarme todo lo que ahora sé, por formarme como profesional y ser humano. Gracias a la UNAM por todo lo que me ha dado, por la formación académica, deportiva y cultural que nos brinda a todos sus alumnos incondicionalmente. Gracias porque los años que pasé en la UNAM han sido los mejores de mi vida y no los olvidaré.

Quiero agradecer a Bettina por apoyarme siempre en las buenas y en las malas, porque sé que siempre puedo contar con ella pase lo que pase; por aguantarme y soportar mis llamadas de atención, por ubicarme cuando pierdo la perspectiva, por ser tolerante conmigo, por esforzarse en este trabajo de Tesis pero, sobre todo, por su amistad. Muchas gracias amiga, te quiero mucho.

Gracias también a mis amigos y compañeros Denisse, Alejandro, Lilia, Elisa, Irased, Ricardo y Liliana, por toda su comprensión, por todo su apoyo, por los buenos momentos que pasamos y por su gran amistad, que espero que dure mucho mucho tiempo más.

Y por último, pero no menos importante, agradezco a la Maestra Gloria Mata por guiarnos a través de la redacción de esta Tesis, por su tolerancia y por apoyarnos hasta el final para obtener nuestro título profesional.

Rita Elizabeth Rodríguez Saavedra

# ÍNDICE

<b>1.</b>	<b>INTRODUCCIÓN</b>	<b>1</b>
1.1	ANTECEDENTES	2
1.2	VIRTUALIZACIÓN	9
1.3	VIRTUALIZACIÓN DE SERVIDORES	13
1.3.1	CONCEPTO	13
1.3.2.	VENTAJAS Y DESVENTAJAS	14
1.3.3.	TÉCNICAS DE VIRTUALIZACIÓN	15
1.3.4.	COMPARACIÓN DE LAS TÉCNICAS DE VIRTUALIZACIÓN	18
<b>2.</b>	<b>RECURSOS PARA LA VIRTUALIZACIÓN DE SERVIDORES</b>	<b>23</b>
2.1	HERRAMIENTAS DE HARDWARE	26
2.1.1	ESPECIFICACIONES DEL SERVIDOR	26
2.1.2.	RECURSOS DEL SERVIDOR	28
2.2.	HERRAMIENTAS DE SOFTWARE	29
2.2.1.	SOLARIS 10	29
2.2.2	GENERADORES DE CARGA	31
2.3.	CONTAINERS	32
2.3.1.	ZONAS	32
2.3.2.	ADMINISTRACIÓN DE RECURSOS	34
<b>3.</b>	<b>DISEÑO DE LA ARQUITECTURA</b>	<b>36</b>
3.1	METODOLOGÍA DEL TRABAJO	37
3.1.1	OTRAS METODOLOGÍAS	39
3.2	METODOLOGÍA DE TRABAJO PARA LA VIRTUALIZACIÓN DE SERVIDORES MEDIANTE SOLARIS CONTAINERS	41
3.2.1	INICIATIVA DE NEGOCIO	41
3.2.2	DIAGNÓSTICO	42
3.2.3	REQUERIMIENTOS Y CARACTERÍSTICAS	49
3.2.4	HERRAMIENTAS	50
3.2.5	DISEÑO Y ARQUITECTURA PROPUESTA	54
<b>4.</b>	<b>IMPLEMENTACIÓN</b>	<b>59</b>
4.1	INSTALACIÓN DEL SISTEMA OPERATIVO BASE	60
4.2.	CONFIGURACIÓN DEL SISTEMA OPERATIVO BASE	64
4.2.1.	RED	65



4.2.2.	HARDENING	67
4.2.3.	CONFIGURACIÓN DE ESPEJO DE DISCOS (RAID 1)	75
4.3.	CONFIGURACIÓN DE SOLARIS CONTAINERS	79
4.3.1.	CONFIGURACIÓN POOL DE RECURSOS	79
4.4.	DEFINICIÓN ZONAS	85
4.5.	INSTALACIÓN DE WEBSTACK	94
4.5.1.	SERVIDOR DE APLICACIONES	95
4.5.2.	SERVIDOR DE BASE DE DATOS	97
4.5.3.	CONFIGURACIÓN WEBSTACK	99
<b>5.</b>	<b><u>EVALUACIÓN</u></b>	<b>103</b>
5.1.	DEFINICIÓN DE PARÁMETROS	104
5.2.	METODOLOGÍA DE EVALUACIÓN	108
5.2.1	PORCENTAJE DE UTILIZACIÓN DE RECURSOS	108
5.2.2.	COSTO TOTAL DE PROPIEDAD O TOTAL COST OF OWNERSHIP (TCO)	111
5.2.3.	ESPACIO FÍSICO	117
5.2.4.	ENERGÍA ELÉCTRICA CONSUMIDA	118
5.3.	OBTENCIÓN Y ANÁLISIS DE RESULTADOS	119
5.3.1	PORCENTAJE DE USO DE RECURSOS	119
5.3.2	COSTO TOTAL DE PROPIEDAD O TOTAL COSTO OF OWNERSHIP (TCO)	130
5.3.3	ESPACIO FÍSICO	141
5.3.4	ENERGÍA ELÉCTRICA CONSUMIDA	143
5.3.5	ENERGÍA TÉRMICA	146
5.4.	PRUEBAS DE FUNCIONALIDAD	148
	<b><u>CONCLUSIONES</u></b>	<b>157</b>
	<b><u>GLOSARIO DE TÉRMINOS</u></b>	<b>161</b>
	<b><u>REFERENCIAS BIBLIOGRÁFICAS</u></b>	<b>164</b>
	<b><u>ANEXOS</u></b>	<b>167</b>
	<b>ANEXO 1 – HOJA DE DATOS SUNFIRE T2000</b>	<b>167</b>
	<b>ANEXO 2 - INSTALACIÓN DEL SISTEMA OPERATIVO BASE</b>	<b>170</b>
	<b>ANEXO 3 - CONFIGURACIÓN DE LOS SOLARIS CONTAINERS</b>	<b>188</b>
	<b>ANEXO 4 – CONFIGURACIÓN DE ASIGNACIÓN DE MEMORIA MEDIANTE MEMORY CAPPING</b>	<b>221</b>

## PRÓLOGO

En la actualidad, las empresas y organizaciones enfrentan nuevas problemáticas que pueden mermar el éxito de sus negocios y operaciones, tal como el costo de la energía y del espacio físico, los cuales son factores que las organizaciones no pueden dejar pasar por alto.

Anteriormente las organizaciones no cuestionaban el comprar un equipo de cómputo para dedicarlo a una única tarea, sin embargo, hay gran interés por aprovechar al máximo los recursos, ya que el no hacerlo se traduce en gastos y pérdidas económicas, incluyendo el tomar en cuenta el consumo mínimo de energía que demanda un equipo, así como el espacio que ocupa el mismo antes de adquirirlo.

Los fabricantes y desarrolladores de tecnologías de información disponen de diversas arquitecturas para la implementación de sistemas informáticos, tal como cliente-servidor, dividido en capas, distribuido, etc., los cuales satisfacen necesidades específicas de manera parcial, ya que aún no existe una que cubra al 100% todas las expectativas, entre las que se encuentran costos, nivel de servicio, facilidad de administración, flexibilidad, escalabilidad, confiabilidad, entre otras.

En los últimos años se han desarrollado diversas tecnologías enfocadas al aprovechamiento máximo de los recursos informáticos, una de ellas es la llamada Virtualización de Servidores en la que mediante la integración de varias cargas de trabajo en un solo equipo físico se evita el tiempo ocioso de los sistemas y reduce el impacto en caso de falla, lo que contribuye a disminuir gastos sin comprometer el nivel de servicio, satisfaciendo así muchas de las expectativas.

La primera implementación de virtualización surge aproximadamente hace 60 años sobre una plataforma mainframe y es hasta comienzos del siglo XXI que surgen diferentes tecnologías de virtualización para otras plataformas, como x86.

Existen diversas técnicas para llevar a cabo la Virtualización de Servidores, como Sistema Operativo, Hypervisor, Paravirtualización, Particionamiento de Hardware, Nivel Kernel, entre otras. En este trabajo de Tesis, nos enfocaremos en una técnica reciente (2005) nativa de Solaris 10 que corresponde a la Virtualización de Sistema Operativo, llamada **Solaris Containers**, que consiste en un conjunto de tecnologías para mejorar la

administración de recursos y aislar ambientes del Sistema Operativo base, teniendo además la ventaja de ser de distribución libre.

Si bien existen algunas implementaciones basadas en la virtualización de servidores, la tendencia es su utilización masiva en los próximos años, de acuerdo con el análisis de especialistas en el área y nuestra propia experiencia.

La Virtualización de Servidores es además, uno de los pilares para el Cloud Computing, un esquema de entrega dinámica de servicios computacionales por medio de la red, probablemente la revolución más importante que actualmente se lleva a cabo, logrando una integración de infraestructura transparente, que facilita las operaciones de las organizaciones.

Por lo anterior, el objetivo preciso de este trabajo de Tesis es diseñar e implementar una arquitectura de virtualización de servidores con la técnica Solaris Containers, para optimizar los recursos de hardware del servidor Sun Fire T2000 y evaluar las ventajas derivadas del diseño.

A lo largo del documento, se lleva de la mano el proceso de implementación. Se incluye también una sección de anexos, donde se detalla paso a paso cada una de las etapas de implementación. En dichos anexos se encuentran las instrucciones, salidas de comandos y capturas de pantalla, con el fin de hacer más comprensible la implementación.

Un capítulo fundamental en este trabajo es el de Evaluación, ya que es en este tema donde se reafirma de manera cuantitativa los resultados de la implementación de virtualización de servidores (ahorro en costos, ahorro de espacio, ahorro de energía, etc), además se aporta un conjunto de metodologías para evaluar proyectos similares, abarcando desde aspectos técnicos hasta financieros.

## CONVENCIONES TIPOGRÁFICAS

La siguiente tabla muestra las distintas tipografías usadas en este documento.

Tipografía	Significado
AaBbCc	Cuerpo del documento
<i>AaBbCc</i>	Anglicismos
AaBbCc	Línea de comandos (información vista en pantalla)
<b>AaBbCc</b>	Marcas y productos registrados
<i>AaBbCc</i>	Descripción breve del capítulo
<b><i>AaBbCc</i></b>	Características destacadas

# CAPÍTULO 1

---

*Dentro del capítulo uno, se describe cronológicamente la evolución de la virtualización, se incluye una línea del tiempo donde se muestra el progreso que ha tenido, también se presenta el concepto de virtualización, su clasificación y un análisis de ventajas y desventajas. Finalmente se describen las diferentes técnicas que existen.*

## **1. INTRODUCCIÓN**

El objetivo de este trabajo es diseñar e implementar una arquitectura de Virtualización de Servidores con la técnica Solaris Containers, con el fin de optimizar el uso de los recursos de hardware del servidor SunFire T2000 y evaluar las ventajas derivadas de esta implementación.

## 1.1 Antecedentes

A finales de los años 50 comenzaron a surgir publicaciones sobre las posibles soluciones para mejorar el aprovechamiento de los equipos de cómputo y satisfacer las necesidades computacionales de la época, pero fue hasta la publicación de Christopher Strachey, el primer profesor de Computación en la Universidad de Oxford y director del Grupo de Programación, titulada “*Time Sharing in Large Fast Computers*” en 1959 donde formalmente se dio a conocer el concepto de Sistema de Tiempo Compartido conocido como *time-sharing* que es uno de los conceptos que precedieron a la virtualización, en esta publicación también se abordaba el tema de la multiprogramación, que es una técnica que permitiría a un programador desarrollar un programa en su propia consola, mientras que otro programador realizaba pruebas, de esta forma se evitaba la espera habitual de los periféricos. La multiprogramación, así como otras ideas innovadoras, comenzaron a impulsar a una nueva serie de equipos.

La Universidad de Manchester desarrolló **Atlas** (Figura 1.1) en 1962 uno de los primeros superordenadores del mundo. Un proyecto precursor en el uso de tiempo compartido y multiprogramación en el que se hacía uso de memoria virtual.

Este equipo introdujo un nuevo componente llamado supervisor el cual se encargaba de gestionar recursos clave, como el tiempo de procesamiento de la computadora y las instrucciones para la ejecución de programas de usuario, el equipo sobresalió por su rapidez de operación. Por otra parte en el año 1965 **IBM** se encontraba trabajando en el equipo **M44/44X**.



Figura 1.1 Equipo Atlas

Este proyecto estaba encabezado por Thomas J. Watson, investigador del Centro Yorktown en Nueva York. La **M44/44X** tenía una arquitectura parecida a la de Atlas, sin embargo en este equipo se introdujo por primera vez el término de máquinas virtuales y

su implementación constaba de una máquina principal IBM 7044 (M44) y de máquinas virtuales 7044 o 44X simuladas de forma parcial. Este fue uno de los grandes aportes a la virtualización por parte de la compañía IBM.

IBM continuó con el desarrollo de nuevos equipos como el **IBM 7094** (Figura 1.2), en este proyecto trabajó en colaboración con el Instituto Tecnológico de Massachusetts (MIT) introduciendo y utilizando la tecnología Sistema de Tiempo Compartido Compatible, (CTSS - *Compatible Time Sharing System*), el término compatible se refiere a que permitía la ejecución por lotes del Sistema Operativo usado en el equipo Fortran Monitor System (FMS), también era capaz de ejecutar una copia de este Sistema Operativo en cada máquina virtual.



Figura 1.2 IBM 7094

En la segunda mitad de los años 60's, MIT e IBM no volvieron a trabajar juntos, cada grupo de trabajo tomo su propio rumbo, MIT continuó en la línea de time-sharing y desarrolló el Proyecto MAC, por su parte IBM centró todos sus esfuerzos en la implementación de un Sistema Operativo para la nueva computadora de **IBM System/360** (Figura 1.3).

Bob Creasy dirigió un equipo que trabajó en este nuevo sistema operativo que incorporó no sólo la memoria virtual, sino también máquinas virtuales, y que más tarde se conoció como **CP-40** este Sistema Operativo fue el precursor de la investigación a **CP-67**. Al sistema operativo en primera instancia se le dio el nombre de **supervisor**, posteriormente con la habilidad de ejecutar sistemas operativos sobre otro sistema operativo, apareció el termino **hypervisor**. También se contaba con un Monitor de Máquina Virtual (VMM - *Virtual Machine Monitor*) el cual se ejecutaba directamente sobre el hardware subyacente, permitiendo múltiples máquinas virtuales (VMs), cada VM podía

ejecutar una instancia de su propio sistema operativo privado al comienzo este era **CMS**, o *Conversational Monitor System*.

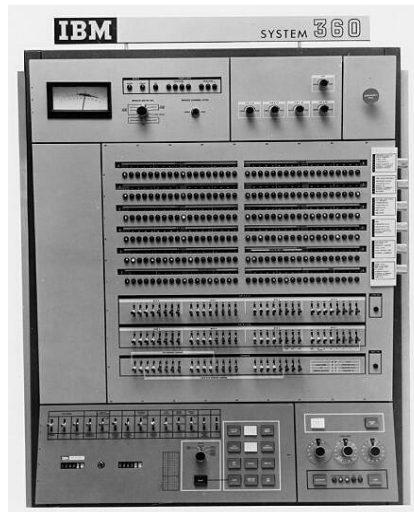


Figura 1.3 IBM System/360

La segunda versión del *hypervisor* de IBM (CP-67) fue desarrollada en 1968, esta nueva versión hacía uso de la compartición de memoria a través de máquinas virtuales, proporcionando a cada usuario su propio espacio de memoria. *hypervisor* admitía tener varios sistemas operativos en un solo hardware, al igual que un número variable de máquinas virtuales ya que en CP-40 sólo se permitía una cantidad fija de catorce entornos virtuales **System/360**, cada uno con 256 KB de memoria virtual. Esta versión se utilizó para la consolidación del hardware físico e implementar más rápidamente ambientes tales como entornos de desarrollo.

En el año de 1972 IBM anuncia el primer sistema operativo, el **VM/370** destinado para la virtualización en los mainframes **System/370** (Figura 1.4).

Era capaz de ejecutar varias máquinas virtuales con más memoria virtual y además del intercambio de información entre las máquinas interconectadas.

Las máquinas virtuales tenían la misma arquitectura que el equipo System/370 y eran gestionadas por un componente llamado un monitor de máquina virtual (VMM) el cual se ejecutaba en el hardware real, cada máquina virtual fue capaz de ejecutar una instalación única del sistema operativo de IBM de manera estable.





Figura 1.4 IBM System/370

**IBM** incorporó mainframes de alto rendimiento, los cuales se volvieron de uso común pero les hacía falta la posibilidad de realizar múltiples tareas, es por esto que desarrollaron un método para crear múltiples particiones lógicas, en estas particiones se ejecutarían varias aplicaciones y procesos al mismo tiempo, que finalmente se les conocieron como VM, Máquinas Virtuales las cuales trabajaban independientemente una de las otras, y cada una utilizando los recursos provistos por el mainframe. Dado que en aquella época los mainframes eran recursos caros, la virtualización permitía aprovechar de una manera eficiente el equipo y con ello recuperar la inversión que se hacía con la compra de estos equipos.

En la década de los 80's, dejando a un lado el uso compartido y centralizado del modelo de mainframe, las organizaciones aprovecharon la llegada de las maquinas **x86**, y comenzaron a implementar sistemas distribuidos; con esto comenzó una nueva era de las micro-computadoras, aplicaciones cliente-servidor, y computación distribuida en donde los enormes y potentes mainframes multitareas de un solo hardware y de tamaño muy grande se comenzaron a cambiar por servidores arquitectura **x86**. No obstante, IBM continuó con el desarrollo de su familia VM y no ha dejado la línea de mainframes.

Los primeros servidores x86 carecían de la potencia necesaria y del procesador adecuado para ejecutar múltiples sistemas operativos, pero permitían la instalación de nuevos Sistemas Operativos como Windows, Unix y Linux, los servidores eran tan baratos y pequeños comparados con los mainframes que las empresas no tardaron en adquirir este hardware y dedicándolo a la implementación de cada aplicación sin cuestionarlo, ya para los años 90 se convirtieron los servidores x86 en el estándar de la industria.

Los profesionales en Tecnologías de la Información (TI) comenzaron a observar problemas relacionados principalmente con el desaprovechamiento del hardware, ya que en las implementaciones en servidores se utiliza sólo una pequeña parte entre un 10% y un 15% de la capacidad total, según señala *International Data Corporation*, un tema que

antes no era tomado en cuenta se convierte en un punto primordial como lo es el consumo energético de los servidores y del sistema de enfriamiento que necesitan, lo que dio lugar a buscar nuevamente en el uso de sistemas compartidos entre varias aplicaciones como la solución a estos y más problemas presentes en ese entonces.

En 1999, **VMware** una empresa fundada en California en 1998 trae al mercado el primer producto para la virtualización, **VMware Workstation**, el cual puede ser implementado en un Sistema Operativo nativo Windows o Linux y se pueden crear máquinas virtuales, las cuales soportan la instalación de un Sistema Operativo “guest” como puede ser Solaris x86, FreeBSD, Windows y Linux. Fue hasta el año 2000 donde VMware incorporó soluciones para servidores con la primera versión VMware GSX Server el cual necesita de un sistema operativo base, ya sea Windows o Linux. Para el siguiente año salió la primera versión de VMware ESX Server, que a diferencia de VMware GSX Server no necesita un sistema operativo base, porque incluye nativamente un *hypervisor*.

Con esta nueva característica, los sistemas operativos y las aplicaciones se aíslan en múltiples servidores virtuales que residen en un único hardware y los recursos del sistema se asignan dinámicamente. Adicionalmente contaba con una consola de administración.

**Xen** fue un proyecto desarrollado por el Grupo de Investigación de Sistemas en la Universidad de Oxford, la versión Xen 1.0 salió al mercado en el año 2003 la cual utilizaba un tipo de virtualización llamado "**paravirtualización**".

Desde sus inicios Xen fue de Distribución Libre, por lo que a lo largo de los años ha tenido contribuciones de empresas como **AMD, Novel, HP, Red Hat** entre otras, resultado de ese esfuerzo conjunto, la versión Enterprise Xen 3.0 se realizó en 2005, esta versión es capaz de soportar hasta 32 procesadores, compatible con la tecnología Intel VT y permite la creación de hasta 32 servidores virtuales.

**Microsoft** se estaba quedando fuera del mercado de la virtualización, es por ello que centro su atención en una empresa llamada Connectix Corporation primero siendo socios, la empresa desarrollaba productos innovadores y los vendían una vez que estos eran populares a empresas reconocidas, en 1997 desarrollaron la primera versión de Virtual PC para una plataforma Macintosh. En esta versión se podía ejecutar una máquina virtual con sistema operativo Windows como “guest”, posteriormente en el año 2003 trabajaron

en un proyecto orientado hacia la virtualización de servidores llamado Connectix Virtual Server, Microsoft les compró los derechos intelectuales al igual que los de Virtual PC.

El 2 de diciembre del 2003 salió la primera versión de Microsoft Virtual PC 2004, aunque se esperaba también el lanzamiento de Microsoft Virtual Server, fue hasta el siguiente año que finalmente que lanzaron Microsoft Virtual Server 2005 con dos versiones, Standard que sólo soporta máximo hasta 4 procesadores y Enterprise que permite hasta 32 procesadores, las máquinas virtuales son creadas y administradas a través una sitio web IIS, posteriormente salió la Virtual Server 2005 R2 Enterprise Edition la cual incluye poder la funcionalidad de tener sistemas operativos Linux como “guest” y la incorporación de soporte para Intel VT (IVT) y AMD Virtualization (AMD-V).

La versión **Solaris 10** salió a la luz en el 2005, esta nueva versión de Solaris tiene como una de sus principales características la tecnología de virtualización a nivel de sistema operativo, llamada **Solaris Containers**.

Para el año 2006 Virtual Server 2005 R2 Enterprise Edition se ofrece como descarga gratuita con la finalidad de poder competir contra VMware quien cambió a GSX Server por VMware Server ofreciéndolo como freeware, esta estrategia realizada por Microsoft tuvo como objetivo hacerse presente en el mercado de la virtualización de servidores e ir ganado terreno. El año de 2007 fue un año de mucha actividad en las empresas y de lanzamientos de nuevas versiones de productos, durante la conferencia anual para profesionales IT VMworld 2007 se anunció a **VMware ESXi** esta nueva versión sustituyó a VMware ESX, la cual presentó muchos problemas de seguridad principalmente relacionadas con su consola de administración, que no fue incluida en esta nueva versión además de que corrigieron estas vulnerabilidades. Por su parte XenSource libera Xen Enterprise v4, versión con mayor estabilidad, resultando en un número ilimitado de máquinas virtuales por servidor físico, otra característica integrada es XenMotion, herramienta para el traslado de una aplicación de software de un servidor a otro sin dejar de ejecutarse. VMware ha tenido una característica similar, llamada VMotion, disponible desde 2003. XenSource se asoció con Symantec para incluir Veritas Storage Foundation, software de gestión de almacenamiento estándar de la industria para backup y recuperación de datos.

Citrix anuncia la adquisición de XenSource el 7 de Agosto del 2007, esta adquisición lleva a Citrix a incursionar en mercado de virtualización de servidores, empresa que antes únicamente ofrecía soluciones de virtualización de aplicaciones.

Microsoft por su parte liberó el Release Candidate de su nuevo producto llamado **Microsoft Hyper-V Server 2008** en Junio de 2007, posteriormente salió la versión oficial disponible en descarga gratuita, se trata de un producto independiente de Microsoft Server 2008 y también salió la versión Windows Server Hyper-V 2008 en la cual además de tener los roles de servidor como DNS, DHCP, WEB, etc. se incluye el rol de Hyper-V para la Virtualización de Servidores. Microsoft incluyó su propio hypervisor, llamado Hyper-V, administra y permite la compartición de recursos entre los servidores virtuales instalados en el hardware, la última versión de estos productos es la R2 que incluye nuevas características como la migración en vivo de servidores, alta disponibilidad mediante clúster, volúmenes compartidos (CSV), además de mejoras de rendimiento y fiabilidad, soporta hardware con hasta 8 procesadores.

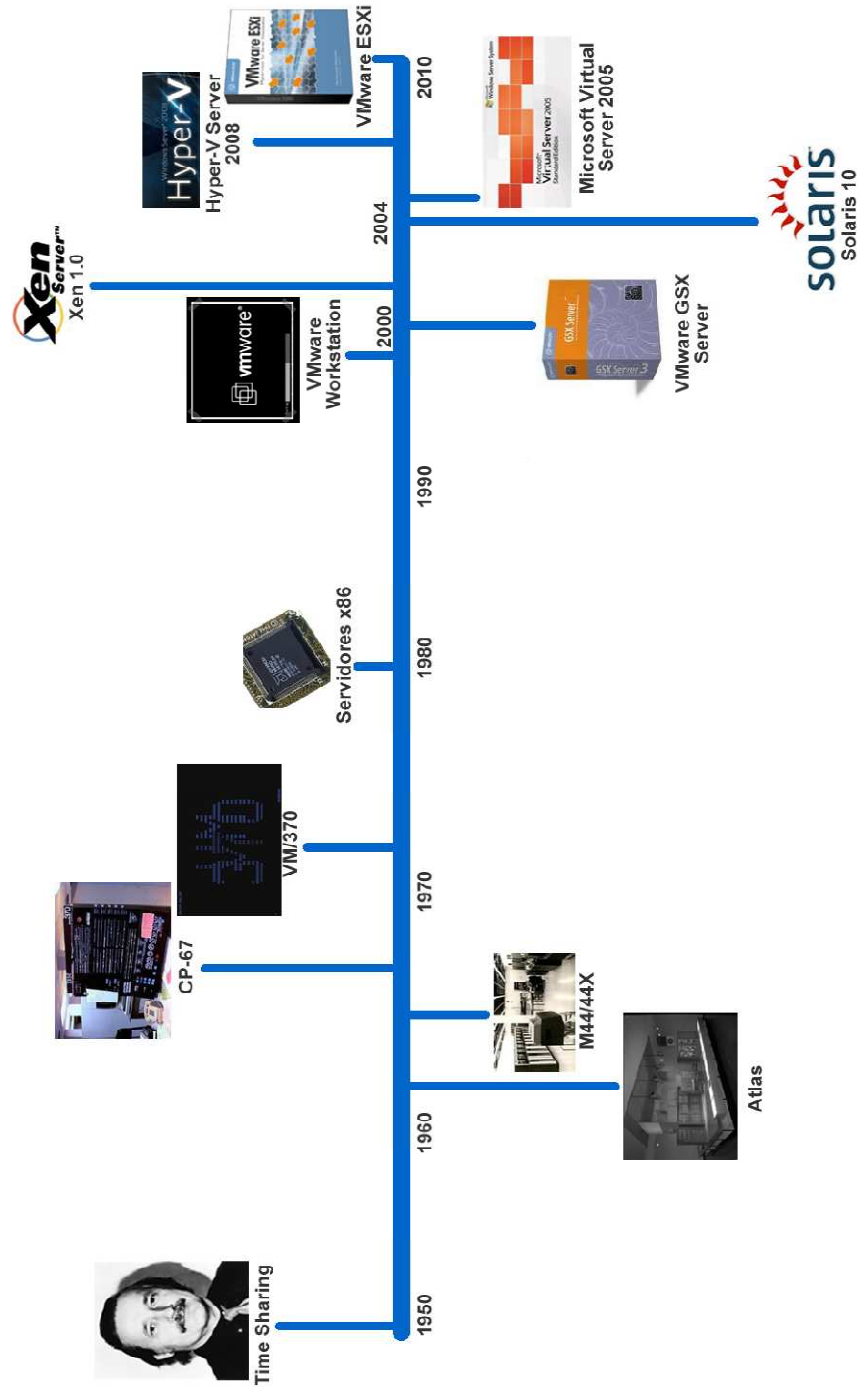


Figura 1.5 Línea de Tiempo de los antecedentes de la Virtualización

## 1.2 Virtualización

La virtualización es una tecnología que permite abstraer recursos físicos con ayuda de ciertas herramientas de software y hardware, estos recursos se utilizan, administran, ejecutan, interactúan, comportan, etc. de la misma forma que un recurso real. La virtualización no sólo se aplica a servidores, se extiende a otros recursos como:

- Virtualización de red
- Virtualización de almacenamiento
- Virtualización de escritorio
- Virtualización de aplicaciones

### Virtualización de red

La virtualización de red es la utilización eficiente de los recursos de la red a través de la segmentación lógica de esta, se simula una red mediante el uso de las diferentes tecnologías que utilizan hardware y software, se gestionan todos servicios en la red como un único grupo de recursos que pueden ser accedidos sin considerar sus componentes físicos.

En la virtualización de red existen dos tecnologías principalmente VPN (*Virtual Private Network*) y VLAN (*Virtual Lan*). Una VPN es una tecnología de red que permite una extensión e interconexión de una red local sobre una red pública o no controlada comúnmente sobre Internet. En ella se crean túneles y se aplican métodos de seguridad para garantizar la integridad de la información entre ambos sitios y poder compartir el acceso a recursos.

Existen diferentes tipo de conexión VPN, la conexión VPN punto a punto se establece entre equipos de hardware dedicados *routers* o *firewalls*, generalmente se utiliza para conectar oficinas remotas con la sede central de la organización y viceversa, en donde se crea un túnel por donde se llevará a cabo la comunicación entre ambos sitios de manera segura

En la conexión VPN de acceso remoto, el equipo remoto debe tener instalado un cliente VPN, por el cual se conecta al servidor de VPN de la organización a través de un túnel encriptado.

Se debe proporcionar credenciales y el servidor VPN se encarga de autenticar, de tal forma que el cliente tenga acceso a todos los recursos como si se encontrar en la red LAN

Una VLAN es la creación de una subred lógica, se define una segmentación lógica basada en el agrupamiento de equipos según determinados criterios, puede ser según los puertos de conexión, la dirección MAC de los equipos, la dirección IP origen de los datagrama o basándose en el protocolo que utiliza, de tal manera que esta segmentación permite un control más inteligente del tráfico de la red, evitando tráfico, aumentando la seguridad y da mayor flexibilidad en la administración.

Existen VLAN Dinámicas (DVLAN) el funcionamiento de estas VLAN's se basa en las direcciones MAC o protocolos utilizados, cuando un equipo se intenta conectarse envía una solicitud con su información, el switch verifica la dirección MAC ingresada previamente por el administrador en la base de datos y automáticamente se configura el puerto al cual corresponde por la configuración de la VLAN. Por otro lado, en una VLAN estática los puertos del switch están ya preasignados a los equipos.

## **Virtualización de aplicaciones**

La virtualización de aplicaciones, permite al usuario ejecutar e interactuar con aplicaciones que no fueron instaladas ni residen físicamente en el equipo cliente, estas aplicaciones ejecutan su propio entorno virtual independiente de otras aplicaciones, se aíslan tanto del sistema operativo como del resto de aplicaciones. La aplicación virtualizada es capaz de guardar cambios de configuración realizados, generándose todo su entorno virtual para que posteriormente se encapsule y se almacene en un servidor, de donde son distribuidas a los equipos clientes.

Los beneficios de esta tecnología son la gestión centralizada de las aplicaciones, así se garantiza la actualización de la versión y la homogeneización, facilitando la administración, además el no consumir recursos en el equipo cliente representa una ventaja, ya que existe un mejor aprovechamiento de recursos.

## **Virtualización de almacenamiento**

La virtualización de almacenamiento tuvo sus inicios con los arreglos RAID, donde un grupo de discos físicos eran para el sistema operativo un grupo de discos lógicos configurables, este concepto sigue prevaleciendo solo que actualmente con otros medios y dispositivos, añadiendo la parte de direccionamiento de red donde la aplicación y usuario no se preocupan por la ubicación física del hardware de almacenamiento.

La información se encuentra centralizada, se tiene mayor tolerancia a fallos, esto es gracias a la incorporación de SAN (Storage Area Network ) que es una red dedicada específicamente a la tarea de transporte de datos para su almacenamiento y recuperación mediante el uso de canales de fibra óptica (Fiber Chanel), en su estructura tiene las conexiones físicas de los sistemas de almacenamiento y los clientes, también se puede utilizar NAS (Network Attached Storage) que diferencia de NAS no tiene una red dedicada.

La virtualización de almacenamiento por software consiste en instalar un software de virtualización en cada host que accede a la red de almacenamiento, este software permite combinar los LUNS/discos virtuales servidos de la red de almacenamiento con total flexibilidad y crear arreglos RAID.

## **Virtualización de escritorio**

Este es un término nuevo para un concepto que no lo es tanto, ya que hace referencia al cómputo personal centralizado, es decir, contar con un sistema central que distribuya la capacidad de cómputo para uso personal, tal cual en las llamadas terminales tontas. En la actualidad la tecnología permite tener la emulación de una PC de Escritorio que en realidad reside en un servidor centralizado, incluyendo toda la funcionalidad que esta puede ofrecer a un usuario final. Con la virtualización de escritorio es posible centralizar el mantenimiento y administración de los escritorios, como aplicar parches de seguridad o para corrección de errores de manera más rápida y confiable, así sólo se aplica una vez y se replica automáticamente en todos los escritorios remotos. Productos que están en esta categoría son Microsoft Remote Desktop Services, Citrix XenDesktop, Sun Virtual Desktop Infrastructure, entre los más destacados.



### **1.3 Virtualización de servidores**

La virtualización de servidores es guiada por la necesidad de consolidar en una misma máquina diferentes aplicaciones y servicios, haciendo más eficiente el uso de los recursos del servidor. La meta de esta tecnología es pasar de administrar servidores individuales con acceso total a los recursos del sistema a administrar un conjunto de recursos de un servidor hacia diferentes aplicaciones.

#### **1.3.1 Concepto**

De forma general, la virtualización se refiere a una forma de crear una versión virtual (no real físicamente) de recursos o dispositivos, como puede ser una computadora, un servidor, un disco duro, una red o un sistema operativo, los cuales son divididos en uno o más entornos de ejecución.

La virtualización de servidores consiste en agrupar dentro de un mismo conjunto de recursos de hardware y software de un servidor, al igual que diferentes aplicaciones y servicios los cuales son implementados en un sistema operativo virtual aislado, también conocido como máquina virtual.

El servidor físico comparte sus recursos, entre los que se encuentran las tarjetas de red, los dispositivos de almacenamiento, la memoria y los procesadores, con las máquinas virtuales creadas en él, por medio de un software que le permite manejar, gestionar, dividir y asignar dinámicamente estos recursos. Gráficamente los recursos se conceptualizan como una pila, en la parte inferior se encuentran los recursos físicos y en la parte superior los virtuales (Figura 1.6). El servidor virtual reconoce los recursos que el servidor físico le comparte como un conjunto de recursos propios y que están bajo el control del servidor virtual.

Con ello, se logra que en un servidor físico existan diferentes máquinas virtuales con diferentes sistemas operativos, diferentes aplicaciones, diferentes cargas de trabajo, diferentes recursos y ser independientes las unas de las otras.



Figura 1.6 Servidor Virtualizado

### 1.3.2. Ventajas y desventajas

La virtualización de servidores ha surgido como una tecnología que permite mitigar algunas de las problemáticas que otro tipo de arquitecturas no cubren, incluso con la ayuda de otros tipos de virtualización, como la virtualización de almacenamiento, virtualización de redes y la virtualización de escritorio, han logrado establecer un punto de diferenciación entre lo que se acostumbraba hacer y la tendencia actual. Entre las principales ventajas de la virtualización de servidores podemos destacar: la reducción del espacio en los centros de datos, la optimización del uso de los recursos del servidor, un ahorro de energía considerable, incluso reducción de costos de operación y soporte, y finalmente hasta un 20% de ahorro en la adquisición de nuevo hardware, según datos de un estudio hecho en organizaciones que han comenzado a cambiar su infraestructura hacia un modelo de virtualización de servidores.

Por otro lado, al ser un conjunto de tecnologías que en fechas recientes está tomando gran importancia para las organizaciones, la principal desventaja es que se requiere un mayor nivel de conocimientos para su implementación, ya que se pueden llegar a maximizar los errores poniendo en riesgo toda la infraestructura y los servicios que en ella se ejecutan, pero esto se debe principalmente al desconocimiento de las consideraciones que una implementación a gran escala puede significar.

Las organizaciones que quieren adoptar la virtualización, incluso los proveedores que atienden a estas organizaciones, no siempre cuentan con una estrategia completa,

asumiendo que únicamente con virtualizar los servidores lograrán mitigar todas las problemáticas que actualmente tienen.

### **1.3.3. Técnicas de virtualización**

Existen diferentes opciones de virtualización de servidores cada una de éstas cuenta con sus propios métodos, herramientas y requisitos, pero siempre bajo el concepto de la virtualización de servidores.

En éste capítulo, presentaremos las Técnicas de Virtualización más populares y actualmente disponibles, así como las características más relevantes de cada una.

#### **Sistema Operativo Host**

Esta técnica consiste en virtualizar un servidor físico a nivel Sistema Operativo, desarrollando múltiples servidores ó ambientes virtualizados, aislados entre sí, ejecutándose sobre un mismo servidor físico.

Este tipo de virtualización yace en un sistema operativo base, el cual actúa como anfitrión para albergar varios sistemas operativos. Cada sistema operativo virtualizado se ejecuta sobre el sistema operativo base, también llamado anfitrión, haciendo uso de recursos compartidos, los cuales son establecidos por el mismo anfitrión.

Los ambientes virtualizados ó máquinas virtuales, son administrados por una aplicación central, que ve a cada ambiente virtual como una aplicación por separado. De esta manera, los ambientes virtualizados asumen que cuentan con sus propios recursos de hardware exclusivos cuando en realidad están compartiéndolos.

Dentro de ésta técnica de virtualización se encuentran VMware Server, VirtualBox y Solaris Containers, entre otros.

VMware Server es una herramienta libre de VMware y está considerada como una solución en ambientes pequeños, ambientes de pruebas o para máquinas personales. Una desventaja de esta herramienta es la limitante de memoria que se le puede asignar a un servidor virtualizado, así como el pobre desempeño en disco.

Solaris Containers es otra herramienta libre que consiste en un entorno separado dentro de una instancia del sistema operativo Solaris. Un "Contenedor", como se le llama

al ambiente virtualizado, permite a los componentes de las aplicaciones aislarse uno de otro aun cuando comparten la misma instancia de sistema operativo. Solaris Containers limita el consumo de los recursos del sistema al mismo tiempo que aísla los diferentes “Contenedores” que habitan en un mismo sistema. Los límites establecidos pueden cambiarse de acuerdo a las necesidades de recursos que van generando las aplicaciones que residen sobre un “Contenedor”.

## **Hypervisor**

El Hypervisor es un software de virtualización que ejecuta un sistema operativo, contrario a la virtualización a nivel de Sistema Operativo que usa un sistema operativo base y corre un software de virtualización como si fuese una aplicación.

La manera en cómo funciona ésta técnica, es que primero se instala el software de Hypervisor sobre el nivel más bajo del equipo físico, el que albergará los servidores virtualizados; después se instala el Sistema Operativo sobre el Hypervisor. El sistema operativo base, es a su vez una máquina virtual, normalmente designada como “*Virtual Machine Zero*” (Máquina Virtual Cero).

Un producto reciente de ésta técnica de virtualización es VMware ESXi, el cual se instala directamente sobre el hardware sin necesidad de una interfaz de sistema operativo sobre el Hypervisor. Sin embargo VMware ESXi debe instalarse sobre un equipo de hardware optimizado y diseñado para la virtualización, ya que la administración de la máquina virtual se realiza a configuración de bajo nivel a través de una interfaz de administración del servidor.

Otros ejemplos de Hypervisor, son por ejemplo Citrix Xen y Microsoft Hyper-V.

## **Paravirtualización**

Es otra técnica de virtualización de servidores que utiliza un software similar al microcódigo de hardware, el cual es el encargado de direccionar los recursos de memoria y dispositivos de servidor hacia las máquinas virtuales, sin enviar las direcciones al procesador.

Como ejemplo de ésta técnica, se encuentran los “Dominios Lógicos”, también conocidos como Logical Domains (LDOMs) de Sun Microsystems, recién adquirida por Oracle. Los Dominios Lógicos permiten crear grupos de recursos, como memoria,

procesador y dispositivos de comunicaciones, que se pueden asignar a múltiples sistemas virtualizados cada uno con su propio sistema operativo y sus propios recursos de sistema.

Otros ejemplos de esta técnica son las herramientas Virtual Iron y Xen, el último es un software de código abierto para sistemas Linux.

### **Particionamiento de Hardware**

Esta técnica consiste en dividir y aislar recursos físicos de un equipo de hardware, de manera eléctrica, es decir, cada grupo aislado de recursos es eléctricamente independiente uno de otro, de tal manera que si un grupo llega a dañarse no daña a los demás.

Todo el hardware involucrado en estos grupos de recursos, como DIMMs de memoria, procesadores, tarjetas de red, tarjetas de fibra óptica, etc., forman un servidor virtualizado independiente.

Ejemplos de esta técnica son los Dominios Dinámicos, de los equipos de alto desempeño de la serie "M" de Sun Microsystems (recién adquirida por Oracle). También se encuentran en esta categoría, los LPARS de los equipos "pSeries" de IBM y los "nPars" de los sistemas Superdome de HP.

### **Nivel Kernel**

La virtualización a nivel de Kernel consiste en que cada máquina virtual ó servidor virtual, usa su propio kernel para arrancar su sistema de archivos independientemente del nivel de kernel del anfitrión.

Dentro de esta técnica encontramos ejemplos como KVM (Kernel Virtual Machine) de Linux y UML (User Mode Linux), el cual usa un kernel ejecutable y la raíz del sistema de archivos para crear un servidor virtual.

### 1.3.4. Comparación de las técnicas de Virtualización

Se muestra en la tabla 1.1 una comparativa entre las diferentes Técnicas de Virtualización

CARACTERÍSTICAS	TÉCNICAS					
	Solaris Containers	OpenVZ	FreeBSD jail	LDom Oracle VM Server for SPARC	Dynamical Domains	Npars
<b>Tecnología de Virtualización</b>	Sistema Operativo Host (Nativo)	Sistema Operativo	Sistema Operativo	Paravirtualización	Particionamiento de Hardware	Particionamiento de Hardware
<b>Plataforma</b>	x86, x64, Sparc	x86, x64, Sparc	x86, x64, Sparc64	UltraSparc T1, T2 T2 Plus, T3	Sparc64	Intel Itanium
<b>S.O. Guest</b>	Solaris8-10 Linux Red Hat	Linux únicamente	FreeBSD únicamente	Solaris, OpenSolaris FreeBSD, Linux	No Aplica	HP-UX, Windows, Linux
<b>S.O. Host</b>	Solaris 10	Linux	FreeBSD Únicamente	Solaris 10	Solaris 10	HP-UX
<b>Tipo de licenciamiento</b>	Libre	Libre	Libre	Libre	Incluido en la compra de Hardware	Incluido en la compra de Hardware
<b>Kernel Compartido</b>	Sí	Sí	Sí	No	No	No
<b>Ambiente Virtual</b>	Contenedor	VP's	Jail	Dominio	Dominio	Npars
<b>Independencia de FileSystem</b>	Sí (modalidad Whole Root Zone)	No	No	Sí	Sí	Sí
<b>Live Migrations</b>	Sí Zona migrations	Sí Vzmigrate	No	Sí	Sí	No
<b>Herramienta de Monitoreo</b>	Comandos de S.O.	Comandos de S.O.	Comandos de S.O.	Comandos de S.O.	Línea de Comandos de S.O.	Global Workload Manager
<b>Asignación dinámica de recursos</b>	Sí	Sí	No	Sí	Sí	Sí
<b>Forma de administración</b>	Línea de Comandos de S.O.	Línea de Comandos de S.O.	Línea de Comandos de S.O.	Logical Domains Manager software	Comandos de System Controller	Global Workload Manager
<b>NIC Exclusiva</b>	Sí	Sí	Sí	Sí	Sí	Sí
<b>NIC Compartida</b>	Sí	Sí	Sí	No	No	No
<b>Granularidad de CPU</b>	VCPU=Thread Físico	CPU		VCPU=Thread Físico	CPU	Celda

CARACTERÍSTICAS	TÉCNICAS				
	KVM	VMware ESX	Vpars	Citrix XenServer	Windows Server 2008 HyperV
<b>Tecnología de Virtualización</b>	Nivel de Kernel	Hypervisor	Hypervisor	Hypervisor	Hypervisor
<b>Plataforma</b>	x86, x64	x86, x64	Intel Itanium	x86,x64	x86,x64
<b>S.O. Host</b>	Linux	No aplica	No aplica	No aplica	Windows Server 2008
<b>S.O. Guest</b>	Linux, Solaris, FreeBSD	Solaris, OpenSolaris, FreeBSD, Linux, Windows, NetWare	HP-UX	Windows, Linux, Solaris	Windows, Linux
<b>Tipo de licenciamiento</b>	Libre	Propietaria, pago por uso	Propietaria, pago por uso	Propietaria, pago por uso	Propietaria, pago por uso
<b>Kernel Compartido</b>	Sí	No	No	No	No
<b>Ambiente Virtual</b>	Máquina Virtual	Máquina Virtual	Partición Virtual	Máquina Virtual	Máquina Virtual
<b>Independencia de FileSystem</b>	Sí	Sí	Sí	Sí	Sí
<b>Live Migrations</b>	Sí	Sí VMotion	Sí	Sí	Sí
<b>Herramienta de Monitoreo</b>	Línea de Comandos	Gráfica, como software adicional con cargo extra	Software de Administración VSE	Gráfica, como software adicional con cargo extra	Gráfica, como software adicional con cargo extra
<b>Asignación dinámica de recursos</b>	No	Sí	Sí	Sí	Sí
<b>Forma de administración</b>	Línea de Comandos	Gráfica, Línea de Comandos	Software de Administración VSE	Gráfica, Consola de Administración	Gráfica, como software adicional con cargo extra
<b>NIC Exclusiva</b>	Sí	Sí	Sí	Sí	Sí
<b>NIC Compartida</b>	Sí	Sí	Sí	Sí	Sí
<b>Granularidad de CPU</b>	CPU	CPU	CPU	CPU	CPU

- **Herramienta:** Nombre comercial de la herramienta, en caso de que aplique, incluye el fabricante
- **Técnica de virtualización:** Técnica usada por la herramienta
- **Plataforma:** Plataformas de CPU en que puede ser ejecutada la herramienta de virtualización
- **S.O. Host:** Sistema Operativo que se ejecuta de forma nativa en el servidor o SO base
- **S.O. Guest:** Sistema Operativo que se ejecuta de forma virtual en el servidor
- **Tipo de licenciamiento:** Tipo de licencia que tiene la herramienta, libre o propietaria.
- **Kernel Compartido:** Capacidad de ejecutar diferentes núcleos de sistema operativo en cada máquina virtual, ya sea por diferencia de S.O. o por nivel de parches
- **Independencia de FileSystem:** Capacidad de operar File Systems o Sistemas de archivos de Sistema Operativo de forma independiente entre máquinas virtuales
- **Live migration.** Capacidad de mover equipos virtuales mientras se siguen ejecutando, entre equipos físicos
- **Herramienta de Monitoreo:** Forma de monitoreo nativo de la máquina virtual, sin uso de software de terceros
- **Asignación Dinámica de Recursos:** Es la capacidad de conceder mayores recursos (red, cpu, memoria) a una Máquina virtual, mientras esta sigue ejecutándose.
- **Administración:** Manera en que se administran las máquinas virtuales.
- **Virtualización de NIC:** Capacidad de virtualizar las interfaces de Red de manera dedicada a la máquina virtual o compartida entre un conjunto de máquinas virtuales
- **Granularidad de CPU:** Nivel más pequeño de administración, asignación y división de los recursos de procesamiento. Existen CPUs físicos, los cuales pueden tener Cores o Núcleos, que a su vez tienen Threads o Hilos de ejecución.



En la tabla 1.1 se describen y comparan las principales características de las herramientas para virtualizar, de donde se observa en primera instancia la técnica de virtualización que utilizan, sobresalen con un mayor número de herramientas la técnica a nivel de Sistema Operativo y la de *Hypervisor*, la primera tiene la ventaja de proveer un mayor rendimiento.

Otras características a tomar en cuenta es la plataforma, **Solaris Containers**, **OpenVZ** y **FreeBSD jail** soportan múltiples plataformas x86, x64 y Sparc, esta propiedad las pone en ventaja sobre otras herramientas, ya que son independientes del hardware lo que conlleva a mayor flexibilidad en la nueva adquisición de un hardware con una plataforma específica y amplía la posibilidad de rehusar hardware con el que ya se cuenta, por otra parte el resto de las herramientas son dependientes de cierta plataforma: Logical Domains a los servidores Sun CMT, Lpars a los sistemas IBM, Vpars a los servidores de HP, y Xen, VMware y Windows Server 2008 HyperV Virtual a los sistemas x86 y x64. En nuestro caso se cuenta con un servidor Sun Fire T2000 que pertenece a la familia de servidores con procesadores UltraSPARC, de la familia Sun CMT, por lo que es posible virtualizar mediante Logical Domains o Solaris Containers.

Los servidores y las aplicaciones requieren de cierto Sistema Operativo para su ejecución y prestación de servicios, las herramientas: Windows Server 2008 HyperV, Npars, OpenVZ y FreeBSD jail no soportan el S.O. Solaris lo que representa un problema de compatibilidad. La herramienta que soporta un mayor número de S.O's es VMware, es capaz de soportar máquinas virtuales con Solaris, Windows, Linux, Netware y OpenSolaris.

Las licencias de las herramientas son un punto crucial, ya que si bien la virtualización tiene beneficios en la reducción de costes, se tienen herramientas que generan gastos en la adquisición de una licencia para su implementación como lo son Windows Server 2008 HyperV, Citrix Xen Server y la más costosa VMware ESXi, estas licencias representan un costo extra asociado a la virtualización. Por otro lado existe Hardware que incluye el software para virtualizar y las licencias asociadas, sin embargo dicho Hardware está categorizado como High-End, teniendo un costo elevado, tal es el caso de Dynamical Domains para los equipos M-Series de Sun o los Npars de los equipos basados en Cellboards de HP. Y es por ello que se tiene preferencia sobre herramientas como **Solaris Containers**, OpenVZ, FreeBSD jail, KVM y LDom que son libres.

**Solaris Containers** y OpenVZ ofrecen el compartir un único kernel de Sistema Operativo con lo cual se obtiene una mejor repartición de los recursos del nivel superior del IT Stack (la base de software sobre la cual se ejecutan las aplicaciones, incluyendo sistema operativo, servidores y lenguajes), así como una disminución significativa de los procesos de traducción, también se reduce la complejidad, puesto que sólo hay un sistema operativo que desplegar y mantener. En las herramientas que utilizan hypervisor, cada kernel de hypervisor es independiente por lo tanto las modificaciones del kernel del hypervisor sólo le afectan a sí mismo y con esto el overhead en los contenedores es más bajo que los que usan hypervisor.

La virtualización de servidores implica una nueva estrategia de administración y monitoreo, herramientas como: Windows 2008 HyperV, VmWare ESXi, Citrix Xen Server y Npars ofrecen soluciones de administración de manera gráfica y con un ambiente “amigable”, sin embargo estas soluciones requiere de una pago extra independiente de la licencia por el uso de dichas herramientas de administración gráfica, este software gráfico cuenta con funciones y comandos ya programados que pueden no ajustarse al nivel de administración específica de una empresa u organización, en contraparte las demás herramientas ofrecen la administración y monitoreo desde la línea de comandos sin ningún costo y con la libertad de ajustar la administración y monitoreo necesario para cada ambiente como es el caso de: **Solaris Containers**, OpenVZ, Ldom y FreeBSD jail.

FreeBSD jail es la única herramienta que no presenta las características de Live migration ni la de asignación dinámica de recursos, dos funcionalidades muy importantes para servidores de Alta Disponibilidad. Otra característica asociada a la asignación dinámica de recursos es el nivel de granularidad, a mayor nivel de granularidad se ve incrementada la asignación y el aprovechamiento de recursos, tomando en cuenta esto, **Solaris Containers**, LDom y Dynamical Domains proveen un mayor nivel.

# Capítulo 2

---

*Dentro del capítulo dos, se definen y especifican los recursos de hardware y software necesarios para la virtualización de servidores mediante la técnica de Solaris Containers, al igual que se explican los principales elementos que se utilizan en esta técnica.*

## **2. RECURSOS PARA LA VIRTUALIZACIÓN DE SERVIDORES**

Haciendo mención al capítulo anterior, existen diversas técnicas de virtualización de servidores, cada una con sus ventajas, desventajas y métodos de aplicación. El presente trabajo de Tesis está enfocado en la técnica de virtualización de servidores a nivel de Sistema Operativo mediante Solaris Containers, la cual elegimos basándonos en diferentes características, entre las que destacan:

- Es de distribución libre
- Aplica para plataformas SPARC y x86/x64
- Puede albergar versiones anteriores a Solaris 10
- Tiene la capacidad de virtualizar ambientes Linux
- Alto nivel de granularidad de uso del procesador

Cuando una organización se encuentre en la necesidad de implementar una solución que involucre la virtualización de servidores, deben de tomarse en cuenta diferentes aspectos para elegir la mejor técnica de virtualización. Dependiendo de las aplicaciones que se piensen virtualizar debe de elegirse la herramienta que mejor satisfaga los requerimientos de las aplicaciones, por ejemplo, algunas aplicaciones funcionan basándose en una versión específica de kernel del Sistema Operativo, para lo cual la mejor herramienta de virtualización sería aquella que cuente con el nivel de kernel aislado entre los diferentes ambientes virtualizados para evitar que alguna actualización del nivel de kernel afecte a los demás servidores virtualizados y que proporcione más independencia a cada ambiente. Algunas de las herramientas que aplicarían para éste tipo de solución son Oracle VM Server for SPARC (antes llamados Sun Logical Domains) y VMWare ESX, entre otros.

La Tabla Comparativa de las diferentes Técnicas de Virtualización presentada en el tema 1.3.4, puede servir como base y ayuda a las áreas de Tecnología de la Información al momento de tomar una decisión para elegir la mejor técnica de virtualización de servidores para sus ambientes.

Así como debe de elegirse la mejor herramienta de software para la virtualización de servidores, también debe de elegirse el mejor equipo de hardware para llevar a cabo la implementación de una solución de virtualización.

El tema de la virtualización de servidores se ha vuelto popular y más organizaciones son atraídas por esta nueva solución, lo cual ha generado que los diferentes proveedores de hardware inviertan en el desarrollo de nuevas tecnologías enfocadas a la virtualización. Tanto AMD como Intel han agregado como características a sus últimos procesadores métodos especiales para que el hardware asista y mejore varios de los procesos que se ven involucrados en la virtualización.

AMD tiene la extensión AMD Virtualization (AMD-V™) mientras que Intel utiliza su denominado Intel® Virtualization Technology (Intel® VT).

Dichas tecnologías permiten crear de manera más eficiente las máquinas virtuales, ayudan a la migración en vivo de máquinas virtuales y conceden el acceso directo de los dispositivos de entrada y salida de una máquina virtual, sin tener la necesidad de pasar por el hypervisor, entre otros beneficios.

En la elección del equipo de hardware deben de tomarse ciertas consideraciones, entre las que destacan:

- La arquitectura del sistema
- El procesador
- Cantidad de bahías para tarjetas PCI-E
- Número de bahías para discos duros
- Soporte a diferentes Sistemas Operativos

Por ejemplo, cuando se decide por una solución de virtualización con la herramienta Oracle VM Server for SPARC (antes llamada Sun Logical Domains) debe de considerarse implementar esta solución en un equipo de hardware que cuente con un procesador **SPARC®** de la tecnología Ultra SPARC CMT® (Chip Multi Threading), ya que éste procesador cuenta con la capacidad de generar varios hilos de ejecución simultáneos para atender varias tareas y la herramienta Oracle VM Server explota estas características al máximo, ya que el diseño de la herramienta está basado directamente en el procesador **Ultra SPARC CMT®**.

Así como se muestra en la Tabla Comparativa del capítulo 1.3.4, existen algunas herramientas de virtualización que sólo pueden implementarse en ciertas arquitecturas, como por ejemplo Windows Server 2008 Hyper-V, la cual sólo puede usarse en plataformas x86/x64 para virtualizar ambientes de la misma plataforma; por el contrario, la herramienta Solaris Containers puede implementarse en plataformas x86/x64 y plataforma SPARC® y puede virtualizar ambientes UNIX y Linux.

Otro punto importante que debe tomarse en cuenta al momento de decidir la adquisición de un equipo de hardware son los costos de software y hardware. Estos son cuantificados usando métodos tales como el Retorno de Inversión (ROI) o el Costo Total de Propiedad (TCO), los cuales son dos mecanismos de negocio importantes que se utilizan para determinar el valor económico de una decisión de compra.

Se pueden encontrar herramientas en línea como “Return of Investment (ROI) Calculator” de Microsoft y el “VMware ROI Calculator” de VMware que permiten simular el escenario y muestra como se obtiene un ROI favorable en la virtualización de servidores, ya que se han hecho varios estudios que indican esto. Para el análisis de TCO, también

existen herramientas que tienen un costo de licencia, por ejemplo “Gartner’s TCO Manager” y “Gartner’s TCO Analyst”, ambos son propiedad intelectual de **Gartner**.

## **2.1 Herramientas de hardware**

En este capítulo, se especifican a detalle los recursos de hardware utilizados en la presente Tesis, tal como lo es el servidor Sun Fire T2000 que pertenece a la familia de servidores con procesadores UltraSPARC®, también llamados CMT (Chip MultiThreading).

El procesador CMT se caracteriza por tener la capacidad de correr varios hilos de ejecución al mismo tiempo, lo cual evita el “tiempo muerto” cuando se ejecuta una operación del procesador y pasa a memoria, es decir, el procesador ejecuta la siguiente tarea encolada sin quedarse ocioso.

Como herramienta de hardware elegimos el servidor Sun Fire T2000, del fabricante Sun Microsystems (recientemente adquirido por Oracle). Este equipo de hardware se eligió debido a que cuenta con tecnología CMT especial para la virtualización, además de que es producto del mismo fabricante de Solaris 10 y a su vez, de Solaris Containers, lo cual representa una ventaja ya que dichas soluciones están adecuadas al hardware del fabricante. Además, la herramienta Solaris Containers es capaz de aprovechar la capacidad de hilos simultáneos de ejecución que posee el procesador Ultra SPARC del servidor Sun Fire T2000.

### **2.1.1 Especificaciones del servidor**

El equipo Sun Fire T2000 cuenta con procesador UltraSPARC® T1, el cual entrega hasta un máximo de 32 hilos simultáneos de ejecución por procesador, lo cual genera el procesamiento ideal no sólo para aplicaciones Web, también para aplicaciones de misión crítica, bases de datos y aplicaciones de Planificación de Recursos Empresariales (Enterprise Resource Planning ERP) y Administración Basada en la Relación con los Clientes (Customer Relationship Management CRM).

Tal como se describe en el documento de especificaciones del equipo Sun Fire T2000 (Ver Anexo 1), el equipo cuenta con capacidades de alta disponibilidad, tales como discos

duros instalables en caliente (“hot plug” en inglés, lo cual significa, que el servidor no tiene que apagarse para instalar un disco duro al chassis), fuentes de poder redundantes y reemplazables en caliente (“hot swappable”, es decir, si una fuente de poder falla, la otra sigue funcionando y la dañada se desmonta del chassis sin apagar el servidor y se instala la nueva sin reiniciar el equipo), monitoreo del ambiente del servidor, manejo de niveles de RAID, entre otras.

El servidor es un equipo de 2 unidades de Rack, el cual entrega un alto desempeño y capacidades de RAS con un mínimo costo de energía en un pequeño espacio, tiene una capacidad máxima de 64GB en RAM, 1 procesador por chassis y Sistema Operativo Solaris 10.

El equipo que usaremos durante el desarrollo de la Tesis, es un equipo Sun Fire T2000 con 1 procesador UltraSPARC® T1 Quad Core a una velocidad de 1.0 GHz, 8 GB en RAM y 2 Discos Duros de 146GB.

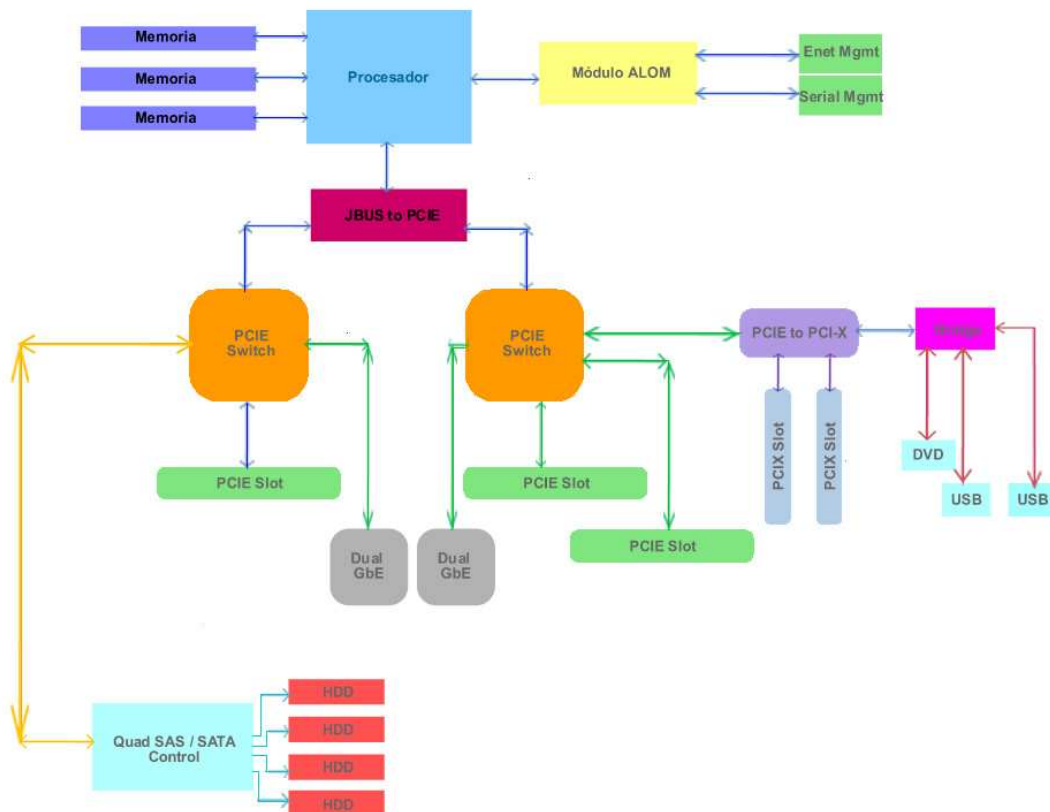


Figura 2.1 Diagrama de Servidor

En la figura 2.1 se muestra la conectividad entre los diferentes componentes que forman la arquitectura del sistema del servidor Sun Fire T2000. Se puede apreciar que los bancos de memoria están conectados directamente al procesador a través de un bus de datos.

También conectado al procesador se encuentra un módulo llamado “ALOM” el cual es un pequeño firmware que permite la administración del servidor a través de los puertos de administración marcados en los recuadros verdes conectados al módulo.

Para la comunicación entre el procesador y las tarjetas de conectividad a la red de datos, se encuentra un bus de datos especial, llamado “JBUS a PCIE”. Este bus es el canal de comunicación entre los switches PCIE y las tarjetas de datos conectadas a ellos. Como se puede ver en el diagrama, existen 5 bahías PCIE, de las cuales 2 bahías están configuradas para soportar tarjetas tipo PCIX; pero para que los switches PCIE reconozcan el microcódigo de éstas tarjetas PCIX, existe un pequeño bus que realiza una conversión de PCIX a PCIE para que la tarjeta sea reconocida por los switches PCIE y a la vez, por el procesador.

Dentro de la arquitectura del procesador existe también un “bridge”, el cual tiene la tarea de comunicar los puertos USB y DVD del chasis hacia el JBUS del procesador.

En cuanto a los discos duros, éstos se comunican al procesador a través de un “Control Quad SAS ó SATA” que parte de los dos Switches PCIE que se conectan al procesador. Este control Quad SAS ó SATA cuenta con 4 bahías de disco duro conectadas para instalar 4 discos duros al servidor.

### **2.1.2. Recursos del Servidor**

El Servidor Sun SPARC T2000 con el que estamos trabajando, cuenta con los siguientes recursos de hardware:

- 2 Tarjetas HBA FC Dual Channel QLogic
- 4 puertos Gigabit Ethernet
- 8 DIMMs DDR2 de 1 GB
- 1 procesador Ultra SPARC T1 con 6 cores q 1.0 GHz
- 2 Discos Duros de 146GB



## 2.2. Herramientas de Software

Las herramientas de software que se utilizan para la implementación y diseño en este trabajo son descritas a continuación, explicando sus principales características y funcionamiento.

### 2.2.1. Solaris 10

Solaris 10 es un sistema operativo UNIX de código abierto lanzado por Sun Microsystems a principios del año 2005, es gratuito, sin importar la cantidad o tipo de sistemas en que se requiera utilizar y compatible con muchas de las arquitecturas de hardware de fabricantes como Dell, Fujitsu, Siemens Computers, IBM y HP. Funciona principalmente en arquitecturas SPARC de 32 y 64 bits, x86, AMD64 y EM64T, al ser un sistema operativo que emplea la técnica de multiprocesamiento con hilos de ejecución, se obtiene un mejor rendimiento de los procesadores.

El sistema operativo Solaris 10 es el software que maneja los recursos y gestiona las tareas que se ejecutan en el sistema, se encarga de los datos almacenados y de la comunicación con los dispositivos de entrada y salida, como monitores, discos duros, impresoras, entre otros. Solaris 10 cuenta con el software ONC+ (*Open Network Computing Remote Procedure Call*), protocolo con el que se proveen varios servicios de red, tales como NFS, NIS. El ambiente de escritorio de Solaris 10, si se cuenta con interfaz gráfica, es un ambiente de ventanas o GUI, con el que se accede a todas las funciones del sistema (Figura 2.2).

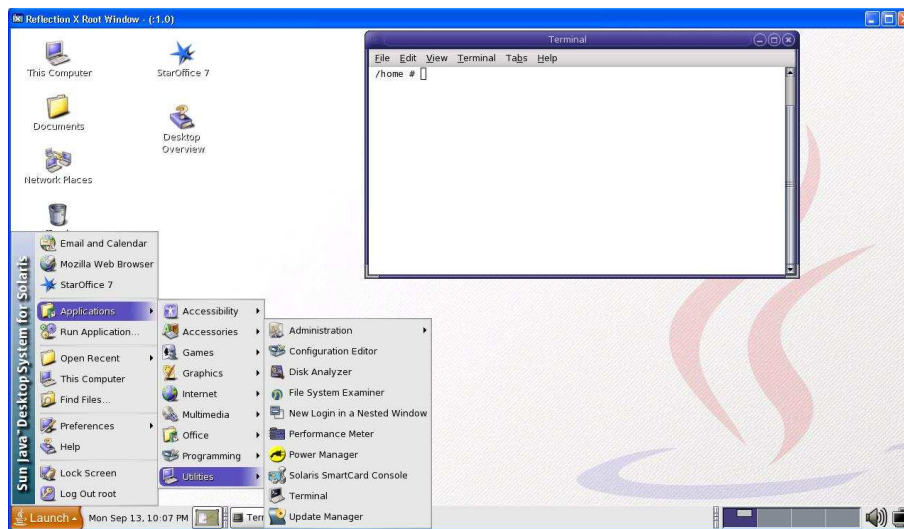


Figura 2.2 Interfaz gráfica de Solaris 10

Los tres principales componentes del sistema operativo Solaris 10 son:

- El **kernel** es el núcleo del sistema operativo, que maneja todos los recursos físicos, entre los que se incluyen: los sistemas de archivos y estructuras de datos, los dispositivos de almacenamiento, las llamadas al CPU, la administración de procesos y el manejo de memoria.
- El **Shell** es la interfaz entre el usuario y el kernel, acepta los comandos introducidos por el usuario, los interpreta y los pasa al kernel para que este los ejecute. Los shells que Solaris 10 principalmente contiene son: Bourne Shell, C Shell y Korn Shell.
- Los **directorios jerárquicos** son la forma en que están organizados el conjunto de archivos del sistema, la estructura tiene forma de árbol con raíz, por lo que la revisión y búsqueda de información se realiza de forma más sencilla, ya que el usuario debe indicar el directorio para localizar el archivo deseado, lo cual se conoce como ruta de acceso del archivo.

Entre las principales características que resaltan de Solaris desde versiones anteriores se encuentran:

- **Seguridad**, proporciona funciones avanzadas que permiten proteger mejor el sistema y consolidar servicios de forma segura, utiliza manejo de permisos de procesos y de usuarios, contenedores y sistemas de archivos. Las aplicaciones y los usuarios sólo son capaces de tener acceso de lectura a los datos, o de escritura, siempre y cuando tenga la autorización apropiada.
- **Rendimiento**, ofrece diferentes ventajas de rendimiento para servicios de base de datos, aplicaciones web y Java. Además de que tiene la capacidad para ejecutarse bajo diferentes cargas de trabajo en sistemas que van desde computadoras de escritorio con un solo procesador hasta sistemas más complejos en los que se requiere el multiprocesamiento.
- **Disponibilidad**, está diseñado para mantener el sistema funcionando en caso de producirse fallas de hardware y durante rutinas de mantenimiento.
- **Facilidad de monitoreo**, las características del kernel combinadas con software desarrollado por usuarios permite monitorear y analizar el comportamiento y rendimiento del sistema.
- **Manejo efectivo** de los recursos de hardware disponibles en el sistema hace posible soportar grandes cargas de trabajo.

La versión Solaris 10 incluye nuevas tecnologías que le permiten distinguirse de otros sistemas operativos existentes, entre las que podemos destacar las siguientes:

- **DTrace** es un conjunto de herramientas de diagnóstico que ayuda a visualizar la actividad que se está ejecutando en el kernel del sistema y las aplicaciones. Los cuellos de botella pueden identificarse y corregirse en el momento que suceden, sin necesidad de hacer pruebas por separado.  
DTrace se ejecuta cuando el administrador necesita recopilar la información necesaria, lo que lo hace una herramienta dinámica ya que al terminar de ejecutarse automáticamente se deshabilita sin afectar el rendimiento del sistema.
- **SMF (Service Management Framework)** es un modelo unificado que provee un conjunto de comandos, utilidades y documentación que facilita el arranque y detención de los servicios que ofrece el sistema operativo.
- **Solaris Containers**, la tecnología de particionamiento de software que habilita la creación y administración de múltiples ambientes virtuales de ejecución sobre la misma instancia de Solaris 10.
- Las tecnologías de **Solaris Fault Manager** y **Solaris Service Manager** incorporan al sistema operativo herramientas para la detección de eventos y proporciona facilidades para la desactivación de componentes con fallas de forma dinámica reduciendo los tiempos de caída y de no disponibilidad.
- **ZFS (Zettabyte File System)** un nuevo sistema de archivos que proporciona una administración más sencilla e integridad en los datos, además de gran escalabilidad al terminar con los problemas asociados al crecimiento de los sistemas de archivos, puesto que cada sistema de archivos consume sólo el espacio que en realidad necesita.

### 2.2.2 Generadores de carga

Los generadores de carga son herramientas de software diseñadas para simular conexiones de usuarios, cargas de trabajo, accesos a dispositivos de almacenamiento y uso de memoria en un determinado momento en el servidor como si fueran reales, con el fin de realizar pruebas de rendimiento para determinar puntos de falla en las aplicaciones, en el hardware y errores de configuración en el sistema operativo.

Simular diferentes cargas de trabajo ayuda a predecir como la aplicación o servicio que brinda el servidor se va a comportar en diferentes niveles de concurrencia.

La **herramienta Stress** permite generar cargas de trabajo o imponer cierto nivel de estrés al procesador, a la memoria y los dispositivos de entrada y salida. La herramienta está desarrollada en el lenguaje C y es de distribución libre, generalmente se utiliza para realizar pruebas de escalabilidad y para exponer errores que solamente se manifiestan cuando el sistema está sometido a una carga de trabajo muy grande. Con la ayuda de esta herramienta se consigue un patrón de utilización de los recursos bajo diferentes escenarios.

La **herramienta JMeter** simula peticiones de diferentes clientes a las aplicaciones de red como HTTP, HTTPS, FTP y realiza pruebas sobre conexiones de bases de datos. Consigue realizar pruebas de funcionalidad y rendimiento, permitiendo la evaluación de las aplicaciones en diferentes cargas de trabajo y bajo distintos niveles de tráfico en la red. Esta herramienta forma parte del proyecto Apache Jakarta.

Existen diferentes herramientas para generar diferentes cargas de trabajo sobre recursos de hardware o aplicaciones, sin embargo, las presentadas en esta Tesis fueron escogidas debido a ser gratuitas y de código abierto. Existen herramientas similares pero tienen un costo de licencia y son recomendadas para pruebas con aplicaciones propietarias, por ejemplo *LoadGen* está diseñada para producir una carga de trabajo en una implementación de Exchange, *LoadRunner* para servidores HP, etc.

## 2.3. Containers

Un Container (Contenedor) en Solaris 10 se define como el ambiente de ejecución que utiliza las tecnologías de Zonas y de Resource Manager. Ambas tecnologías permiten aprovechar distintas ventajas de la virtualización, complementándose entre sí, mientras las Zonas proveen de un ambiente virtual aislado, Resource Manager permite controlar y administrar los recursos de cómputo destinados a cada Zona, dando como resultado un mejor aprovechamiento de los recursos físicos sin comprometer el nivel de servicio.

### 2.3.1. Zonas

Como se ha mencionado hasta ahora, la virtualización consiste en ejecutar dos o más instancias de sistema operativo en un mismo equipo físico, en Solaris 10 la tecnología de particionamiento que permite hacer esto es llamada Solaris Zones, o simplemente Zonas.

Dichas zonas consisten en un conjunto de procesos que permiten crear ambientes de ejecución aislados lógicamente. De esta forma cada ambiente se administra de forma independiente, sin afectar aplicaciones ejecutándose en otras Zonas.

Las Zonas separan las aplicaciones y los datos contenidos en ellas del Hardware en donde residen, facilitando la movilidad entre Hardware. Cada Zona posee sus propios recursos lógicos de sistema operativo, como si fuera una instancia instalada en un equipo físico, tales como passwords de administrador, usuarios, puertos de red, sistemas de archivos, procesos y servicios, interactuando únicamente con sus recursos locales (Figura 2.3).

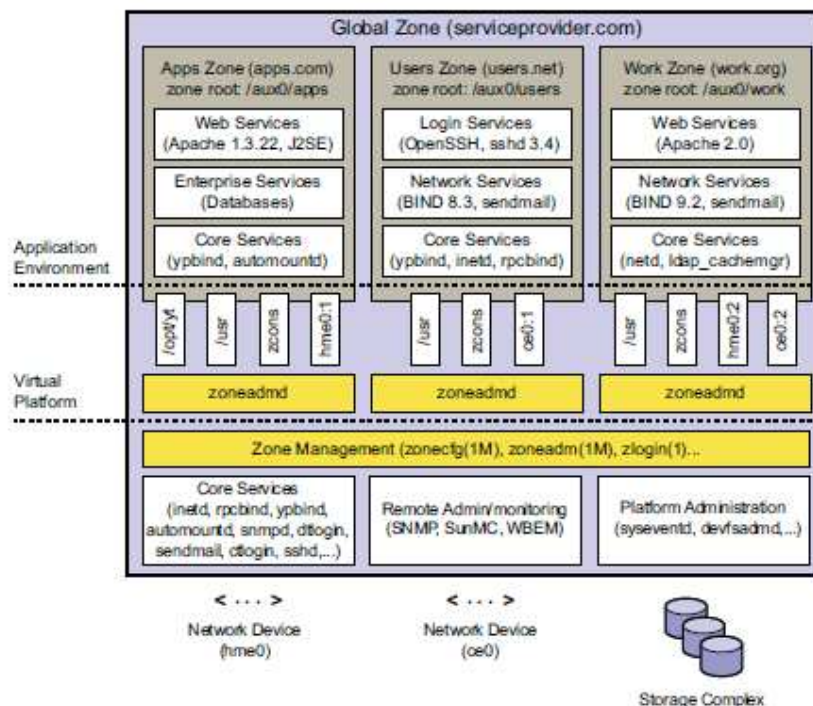


Figura 2.3 Sistema con Zonas Instaladas

Los recursos físicos son compartidos por todos los ambientes virtuales, conocidos como Zonas Locales, mientras que el sistema operativo nativo, llamado Zona Global, se encarga de administrar el uso y acceso de esos recursos, también es la que permite crear, instalar y administrar las Zonas Locales.

Las Zonas Locales contienen un subconjunto de los paquetes del sistema operativo Solaris, además de software adicional o compartido con la Zona Global, así como archivos, datos y configuraciones, sin embargo comparte la operación del Kernel con la Zona Global.

A diferencia de otras técnicas de virtualización de servidores, Solaris Zones tiene dos modelos para instalar el ambiente virtual, el llamado Spares Root Zone, que consiste en compartir la mayoría de archivos binarios con la Zona Global y las demás Zonas Locales, y el Whole Root Zone, semejante a las soluciones de otros fabricantes, en donde los archivos binarios son independientes entre un ambiente virtualizado y otro.

Así como existen dos modelos para instalar las Zonas Locales, existen dos tipos de configuraciones para la red, Shared-IP (IP compartida) y Exclusive-IP (IP exclusiva). En ambos casos existe una virtualización de las interfaces de red, sin embargo, para la configuración de IP compartida la interfaz física de red se comparte con la Zona Global y eventualmente con las otras Zonas Locales, mientras que la configuración de IP exclusiva utiliza una interfaz de red de manera que ni la Zona Global ni otras Zonas Locales puedan hacer uso de dicha interfaz.

Estas dos últimas características, el modelo de instalación del ambiente virtual y la configuración de la red, son elementos clave que se deben considerar al momento de planificar la virtualización, ya que por un lado pueden ofrecer flexibilidad y consolidación de recursos, mientras que por otro lado, de manera excluyente, pueden permitir una mayor independencia entre recursos de los ambientes virtualizados.

### **2.3.2. Administración de Recursos**

Resource Management (administración de recursos) en pocas palabras se refiere a la habilidad de controlar los recursos disponibles que las aplicaciones pueden usar en un sistema, donde básicamente estos recursos son procesamiento y memoria. El Solaris Resource Manager es una tecnología integrada de manera nativa en Solaris 10, lo que significa una mejor integración con otras herramientas. Resource Manager permite reservar recursos de cómputo para una aplicación o proceso en particular, saber cómo están siendo usados los recursos y ajustarlos cuando sea necesario.

Gracias al uso de Resource Manager es posible modificar el comportamiento del sistema operativo respecto a diferentes cargas de trabajo, optimizando así el uso de los recursos, mejorando el rendimiento que se refleja en un mejor tiempo de respuesta y en un mejor servicio.

Existen tres mecanismos para el control de la administración de recursos en el sistema operativo Solaris: mecanismos de restricción, mecanismos de planificación y mecanismos de particionamiento.

Los **mecanismos de restricción** permiten definir los límites de consumo de recursos específicos para una carga de trabajo. Este mecanismo es usado para controlar cargas cuyo comportamiento incorrecto comprometa el nivel de servicio de otras aplicaciones.

Por otro lado, los **mecanismos de planificación** hacen una toma de decisión sobre la reserva y la asignación de los recursos basados en intervalos, así si un recurso está reservado para una aplicación, pero en un instante no está siendo usado, otra aplicación puede hacer uso de él. Solaris 10 Resource Manager hace uso de Fair Share Scheduler (Planificador de Reparto Justo) o FSS para administrar los recursos de procesamiento bajo este mecanismo.

Finalmente, el **mecanismo de particionamiento** describe el uso de subconjuntos de recursos destinados a una aplicación o carga de trabajo, de esta forma cada aplicación asignada a un subconjunto de recursos, dispondrá de un mínimo asegurado, sin embargo, no permite el uso de los recursos ociosos que pertenezcan a otro subconjunto.

# Capítulo 3

---

*En este capítulo se explica la metodología utilizada en el proyecto de virtualización de servidores mediante Solaris Containers y se detalla cada una de sus etapas; así mismo se realiza una comparación con otras metodologías. Se explican detalladamente las características y requerimientos de la arquitectura y del diseño.*

## **3. DISEÑO DE LA ARQUITECTURA**

Mucho hemos mencionado de los beneficios de la virtualización de servidores, sin embargo, llevar a cabo una implementación suele ser más compleja que una *prueba de concepto*, ya que existen diversos factores que se deben considerar. Definir una buena arquitectura de sistemas no sólo depende del ámbito de los servidores, sino de todos los elementos que con él interactúan, como la red, el almacenamiento y los que consideramos más importantes son los usuarios, los objetivos y estrategias de negocio, pues son estos los que definen las directrices, los alcances y los criterios que han de tomarse en cuenta.



### 3.1 Metodología del trabajo

Llevar a cabo un proyecto de virtualización debe considerarse como parte de una estrategia de Tecnologías de la Información (TI), por ello alinearse a una metodología, ayuda a lograr los objetivos como los mencionados en el capítulo 1, reduciendo los riesgos y evitando un gasto mayor.

En este trabajo nos apegaremos a una **metodología general de proyectos**, ya que ésta es independiente de algún proveedor y engloba las etapas de otras establecidas por fabricantes como HP, Sun Microsystems o IBM., sin embargo es importante mencionar que todas tienen etapas en común, por ejemplo Sun Microsystems le llama a la etapa inicial Estudio de Factibilidad mientras que IBM le llama Previrtualización, siendo que la etapa que engloba ambas se llama Iniciativa del Negocio.

En este subtema explicamos cada una de las etapas que integran la metodología de trabajo en la que nos basamos.

La **metodología general de proyectos** contempla 5 etapas secuenciales, las cuales se presentan en la figura 3.1:

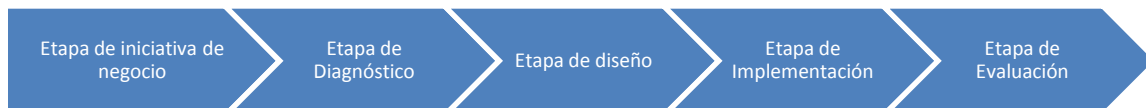


Figura 3.1 Etapas de la Metodología General de Proyectos

**Iniciativa de negocio.-** Esta etapa, común a casi cualquier proyecto dentro de una empresa, es el motivo inicial para que se lleve a cabo algún cambio, mejora o implementación dentro de la organización. Es necesario hacer un estudio de factibilidad que determine si es viable el cambio a una arquitectura de servidores virtualizados. Se consideran tres aspectos básicamente que serán involucrados en el proyecto:

- Los objetivos de negocio
- Los criterios de éxito
- Los servicios

Como ejemplo de los objetivos de negocio están, entre los más comunes, la reducción del costo de mantenimiento, reducción del costo de adquisición, aumento de los niveles de servicio, aumentar los niveles de uso de recursos.

Los criterios de éxito generalmente están asociados con indicadores y métricas que se derivan de los objetivos de negocio. Por último, listar los servicios que estarán involucrados dará una idea más clara sobre el alcance que se tiene.

**Diagnóstico.-** Esta etapa es medular, ya que en ella se realiza la recolección y el análisis de la información de los servidores que serán virtualizados, ya sean servidores que cambian de servidor físico a virtual o servidores que desde su nacimiento serán virtuales. La recolección de datos se obtiene de las aplicaciones, servidores, dispositivos de almacenamiento, red, seguridad e infraestructura en el centro de datos. En cuanto al servidor que será virtualizado a partir de uno físico, es importante tener métricas de uso, equivalencias en caso de que la arquitectura del servidor (CPU, Memoria) sea diferente entre uno y otro servidor, es decir, entre el servidor actual y el servidor que será usado para la virtualización. Para ambos casos, de físico a virtual y virtualizado desde su concepción, es necesario corroborar la compatibilidad y el soporte bajo ambientes virtuales del software que ha de ejecutarse, para que el nivel de servicio no sea afectado.

**Diseño.-** Esta etapa retoma la documentación elaborada en la etapa anterior, así como las reglas de negocio, políticas de seguridad, *mejores prácticas* y la experiencia en otras ramas de sistemas informáticos y tecnologías de la información, todo esto será para dar como resultado la guía sobre la cual se implementará, así como los procedimientos que han de seguirse. Es importante resaltar que esta etapa es iterativa, en la gran mayoría de los casos, una única aproximación no es suficiente para cubrir de la mejor forma todos los aspectos y directrices que se establecieron en la primera etapa (iniciativa de negocio). El Diseño consiste en la distribución de cargas de trabajo en diversos servidores físicos, así como en el dimensionamiento de los recursos necesarios para cada uno de los ambientes virtuales, incluso la definición de esquemas de alta disponibilidad tienen que especificarse en esta etapa. Por último, se documentan nuevos procedimientos para la operación de esos ambientes virtuales.

**Implementación.-** Es la etapa en que los servidores y demás recursos informáticos son instalados, configurados y optimizados (es común el uso del término *tunning*), de

acuerdo al diseño realizado. Puede incluir la adquisición de servidores, aplicaciones, licencias, dispositivos de red, así como de modificaciones al centro de datos.

***Evaluación.-*** Posterior a la implementación, se deben realizar pruebas para validar si se cumplieron los requerimientos especificados en el diseño, así como pruebas de funcionalidad, para dar paso al ambiente operativo, que será el que esté otorgando el servicio a los usuarios. Es en esta etapa cuando se comienza a obtener métricas para determinar el éxito de la implementación.

### **3.1.1 Otras Metodologías**

Se incluye la tabla comparativa de las diferentes metodologías (tabla 3.1), en la cual, en base a la metodología general (que abarca 5 etapas) se engloban las de Sun Microsystems, IBM y HP respectivamente, cabe mencionar que dichas metodologías particulares son propietarias y no hay mucha información pública acerca de ellas.

La tabla muestra que en la etapa referente a “Diagnóstico”, la metodología de IBM no cuenta con ella, aunque es probable que sí se lleve a cabo pero no como un proceso formal; pasa lo mismo con la etapa “Iniciativa de Negocio” no contemplada formalmente por HP, esta etapa es de suma importancia en la toma de decisiones de un proyecto de Virtualización, por lo que seguramente debe de llevarse a cabo dentro de la metodología formal del proveedor, pero no encontramos información que lo corrobore.

**Tabla 3.1 Comparativa de metodologías**

Metodología General de Proyectos: Tesis	Sun Microsystems	IBM	HP
<b>Iniciativa del Negocio</b>	Estudio de Factibilidad <ul style="list-style-type: none"> <li>Objetivo de negocio</li> <li>Criterio de éxito</li> <li>Servicios objetivo (servidores y aplicaciones)</li> </ul>	Previrtualización <ul style="list-style-type: none"> <li>Envolvimiento en virtualización</li> <li>Evaluar el ambiente actual de TI (End-to-End)</li> <li>Determinar si la virtualización es apropiada para el negocio</li> </ul> Hacer pruebas de concepto	
<b>Diagnóstico</b>	Diagnóstico <ul style="list-style-type: none"> <li>Aplicaciones</li> <li>Servidores</li> <li>Almacenamiento</li> <li>Red</li> <li>Seguridad</li> <li>Infraestructura</li> </ul>		Determinar los sistemas a consolidarse <ul style="list-style-type: none"> <li>Capacidades y cargas de trabajo</li> <li>Capacidades de conexión a LAN y SAN</li> <li>Seguridad necesaria en cada uno de los sistemas a consolidarse</li> <li>Colisiones de propiedad de la aplicación</li> <li>Licenciamiento</li> <li>Niveles de Servicio</li> </ul>
<b>Diseño</b>	(Diseño de la) Arquitectura <ul style="list-style-type: none"> <li>Aplicaciones</li> <li>Servidores destino</li> <li>Almacenamiento</li> <li>Red</li> <li>Seguridad</li> <li>Infraestructura</li> </ul>	Planeación y diseño <ul style="list-style-type: none"> <li>Aseguramiento de compatibilidad</li> <li>Dimensionamiento de ambientes virtuales</li> <li>Plan de crecimiento</li> </ul> Optimización de ambientes virtuales	Crear marco hipotético <ul style="list-style-type: none"> <li>Este marco refleja la consolidación de los servidores</li> </ul> Modificar marco hipotético <ul style="list-style-type: none"> <li>El marco es susceptible a mejoras de forma iterativa</li> </ul>
<b>Implementación</b>	Implementación <ul style="list-style-type: none"> <li>Aplicación</li> <li>Servidores</li> <li>Almacenamiento</li> <li>Red</li> <li>Seguridad</li> <li>Infraestructura</li> </ul>	Implementación Implementación de lo planificado, que incluso puede variar de acuerdo a la optimización	Implementación
<b>Evaluación</b>	Administración <ul style="list-style-type: none"> <li>Administración empresarial</li> <li>Monitoreo</li> <li>Optimización</li> <li>Infraestructura</li> <li>Seguridad</li> </ul>	Administración <ul style="list-style-type: none"> <li>Estrategia de mantenimiento</li> <li>Monitoreo y tuneo del sistema virtual</li> </ul>	Administración

### 3.2 Metodología de trabajo para la virtualización de servidores mediante Solaris Containers

Enseguida, se detalla la metodología antes descrita, con un escenario típico en las organizaciones, el que será el implementado en este trabajo.

#### 3.2.1 Iniciativa de negocio

Como se ha mencionado anteriormente, los sistemas informáticos, en particular las tecnologías de la información, han formado parte de la vida contemporánea y han evolucionado de tal forma que no imaginaríamos muchas actividades sin ellos. En nuestro caso, la infraestructura que se ha considerado, servirá para ejecutar aplicaciones enfocadas a 3 de los aspectos que hoy en día parecieran indispensables en una organización lucrativa, sobre todo del rango de PyME: un manejador de contenidos, un comercio electrónico (*e-commerce*) y un sitio de colaboración, siendo cada uno de ellos indispensable para una función específica dentro de la organización.

Como objetivos de negocio, la propuesta debe resultar en una reducción del costo de mantenimiento, aumento en los niveles de uso de los recursos y mínimo, mantener el mismo nivel de servicio. Los sistemas que serán incluidos en el alcance de este trabajo son los Servidores de Aplicaciones (*Application Servers*) y los Servidores de Base de Datos (*DB Servers*) que están destinados a los servicios antes mencionados (*manejador de contenidos, e-commerce y sitio de colaboración*).

En cuanto a descripción e importancia, cada una de las aplicaciones mencionadas, consisten en lo siguiente:

- **Manejador de Contenidos (CMS)** es un sistema integral para la administración de publicaciones electrónicas, con este tipo de software es posible crear, editar, publicar y manipular contenido, ya sea de manera unilateral (ejemplo: una revista electrónica) o de manera colectiva (foros y blogs). Este software es usado por las empresas para publicar nuevos productos, promociones, campañas publicitarias y en general cualquier comunicación que quiera hacer hacia el público, incluso, de manera interna se emplean como herramienta de comunicación hacia el personal, aprovechando sus capacidades de difusión electrónica.

- **Comercio Electrónico**, o sistema de gestión de comercio electrónico, es un conjunto de programas que permiten administrar las ventas en línea y dependiendo de la complejidad con que haya sido concebido, auxilia en labores desde llevar el inventario, hasta el seguimiento de la transacción, asimismo facilita las labores financieras y contables, sin embargo, su principal función es la de ser una interfaz hacia el consumidor, quien simulando estar en una tienda de autoservicio, elige de entre los productos y servicios ofertados aquellos que son de su interés.
- Los **sitios de colaboración** son una herramienta de software muy importante para empresas en donde la interacción entre áreas o personal es de vital importancia para la realización de las actividades propias de la empresa. El concepto puede ser semejante al de un manejador de contenidos, sin embargo, a diferencia de estos últimos, los sitios de colaboración no tienen como objetivo difundir un mensaje, sino servir como repositorios de conocimiento, documentación y productos digitales, tales como imágenes o videos. Los sitios de colaboración son ampliamente usados en *organizaciones proyectizadas* (es decir, aquellas organizaciones que basan su operación en la realización de proyectos).

Si bien, cada una de estas aplicaciones requeriría un proceso complejo de desarrollo para que se ajuste a cada organización, en este trabajo, dado que no es el punto medular, haremos uso de software que ha sido desarrollado de manera genérica bajo el concepto de *Open Source* o Software Libre, lo que representa muchas ventajas, ya que ha sido creado por una comunidad de desarrolladores profesionales, puesto a prueba en implementaciones alrededor del mundo y sobre todo que es de distribución libre y se enriquece de las aportaciones que la misma comunidad realiza al código.

### 3.2.2 Diagnóstico

En la etapa de Iniciativa de Negocio ya se han establecido los alcances generales, es decir, aquellos servicios que se desea optimizar, sin embargo, es necesario entrar a mayor detalle en los aspectos técnicos que cada uno de esos servicios involucran. Al final de este análisis se presenta un diagrama llamado *Mapa End-To-End*, en donde se muestra la interacción de todos los servicios.

- Aplicaciones: Como se comentó en el punto anterior, las aplicaciones son **Código Abierto**, de tal forma que mostramos los requerimientos y las características que los desarrolladores han especificado para cada una, enlistamos los requerimientos de software, los diagramas de flujo de información y el repositorio de donde se obtuvo el código.
  - Manejador de contenidos (CMS):
    - Nombre: *Joomla*
    - Versión: 1.5.20
    - Repositorio: <http://www.joomla.org/download.html>

La tabla 3.2 muestra los requerimientos de software.

Software	Versión				Repositorio
	Mínima	Recomendada (Documentación)	Óptima	Última Versión	
PHP	4.3.X	4.4.7	5.0.X	5.3.3	<a href="http://php.net">http://php.net</a>
Apache	1.3-	1.3.9	2.0.61	2.2.16	<a href="http://httpd.apache.com">http://httpd.apache.com</a>
MySQL	3.23.x -	4.1.22	5.0.X	5.1.51	<a href="http://www.mysql.com">http://www.mysql.com</a>

La figura 3.2 muestra el flujo de información y la interacción que tienen los servidores requeridos.



Figura 3.2 Flujo de Información Manejador de Contenidos

- Comercio electrónico (e-commerce):
  - Nombre: *Zen Cart*
  - Versión: 1.3.9g
  - Repositorio: <http://www.zen-cart.com/>

La tabla 3.3 muestra los requerimientos de software.

Software	Versión				Repositorio
	Mínima	Recomendada (Documentación)	Óptima	Última Versión	
PHP	-	PHP5	-	5.3.3	<a href="http://php.net">http://php.net</a>
Apache	-	Apache2	-	2.2.16	<a href="http://httpd.apache.com">http://httpd.apache.com</a>
MySQL	-	MySQL5	-	5.1.51	<a href="http://www.mysql.com">http://www.mysql.com</a>

La figura 3.3 muestra el flujo de información y la interacción que tienen los servidores requeridos para Comercio Electrónico.

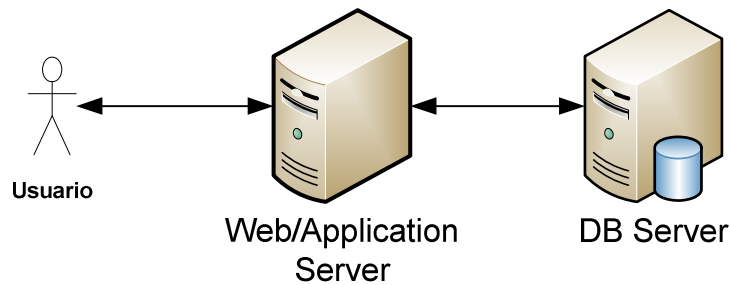


Figura 3.3 Flujo de Información Comercio Electrónico

- Sitio de Colaboración:
  - Nombre: *Collabtive*
  - Versión: 064
  - Repositorio: <http://collabtive.o-dyn.de/>

La tabla 3.4 muestra los requerimientos de software.

Software	Versión				Repositorio
	Mínima	Recomendada (Documentación)	Óptima	Última Versión	
PHP	-	5.1	-	5.3.3	<a href="http://php.net">http://php.net</a>
Apache	-	Apache2	-	2.2.16	<a href="http://httpd.apache.com">http://httpd.apache.com</a>
MySQL	4.1	5.x	-	5.1.51	<a href="http://www.mysql.com">http://www.mysql.com</a>



A continuación (figura 3.4) se muestra el flujo de información y la interacción que tienen los servidores requeridos para el Sitio de Colaboración.

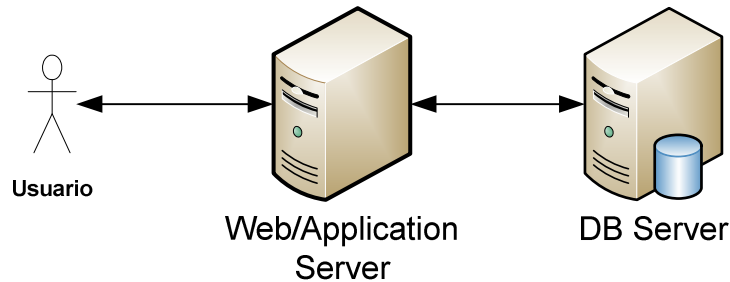


Figura 3.4 Flujo de Información Sitio de Colaboración

- **Plataforma**

El estado inicial de los servidores que estaban destinados a las aplicaciones antes descritas fueron implementados de forma dedicada, conteniendo el software necesario para proveer la capa web, la capa aplicativa y la capa de base de datos, esto se conoce como un servidor monolítico.

La configuración de un servidor monolítico representa ciertas desventajas de seguridad, administrativas y de rendimiento. Con respecto al rendimiento, al tener los recursos compartidos entre las distintas cargas de trabajo pueden estar siendo consumidos en mayor parte por una de ellas, típicamente las bases de datos, reduciendo el tiempo de respuesta del servicio en forma general. Por otro lado, la seguridad se ve mermada, ya que no es posible aplicar las políticas de acuerdo al tipo de carga de trabajo, si bien el web server puede ser accesado por los usuarios, la base de datos debe estar protegida, evitando inconsistencias y datos corruptos. Por último, administrativamente es complejo realizar mantenimientos a un software sin afectar al servicio completo.

A continuación presentamos un diagrama por cada servicio (figura 3.5, 3.6 y 3.7), en los cuales se muestra la forma en que los servidores están siendo usados. En cada diagrama se describe el servidor físico, el sistema operativo y dentro de recuadros el software que en él se ejecuta, por último, las flechas indican la interacción entre cada software.

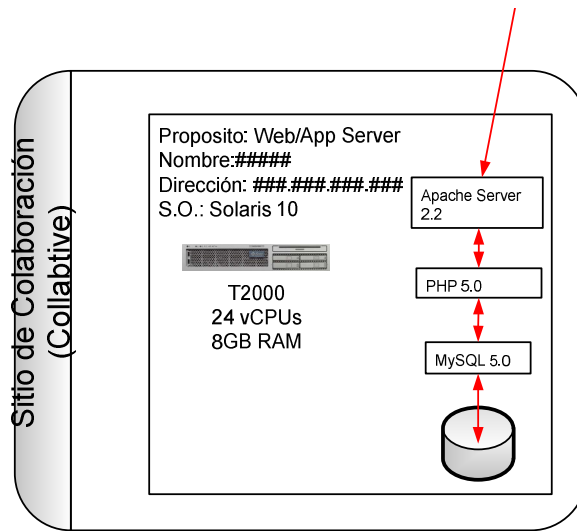


Figura 3.5 Plataforma para servicio de E-Commerce

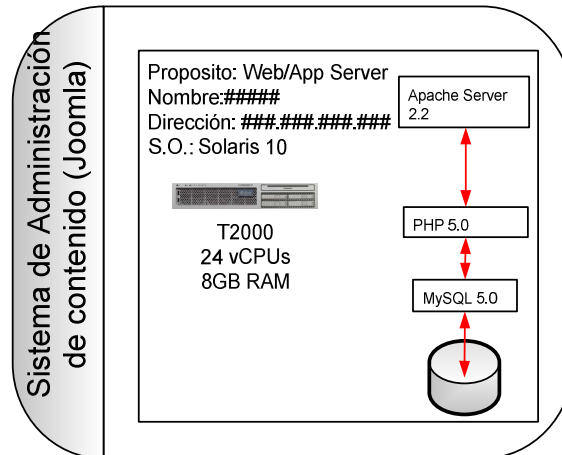


Figura 3.6 Plataforma para servicio CMS

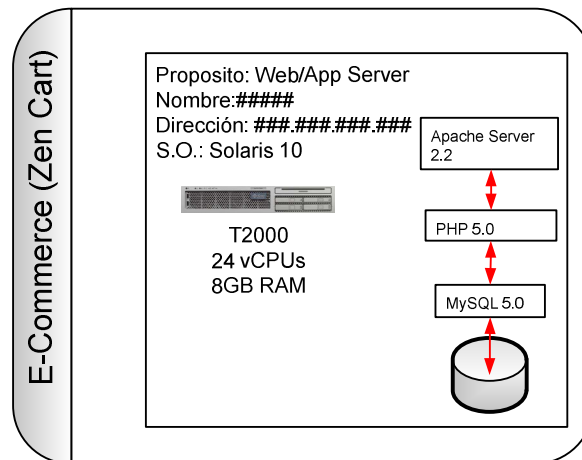


Figura 3.7 Plataforma para servicio de Colaboración

- **Almacenamiento**

Cada servidor cuenta con un disco interno de 146 GB donde reside el sistema operativo, los aplicativos y los datos. Los datos, que es la parte más valiosa de un sistema informático, residen en un dispositivo susceptible a más fallas que otros, debido a que es un dispositivo mecánico. El peligro que se corre al tener de esta forma el almacenamiento, es que ante una falla, es posible perder toda la información, en ocasiones imposible de recuperar.

- **Red**

Cada servidor cuenta con una conexión a la red, sea pública o a la red privada. En caso de que esta conexión falle, el servicio se verá interrumpido.

Por último en la etapa de Diagnóstico, presentamos un diagrama que representa la arquitectura inicial de los servicios y su interacción, este diagrama es conocido en la industria como *Mapa End-To-End* (figura 3.8).

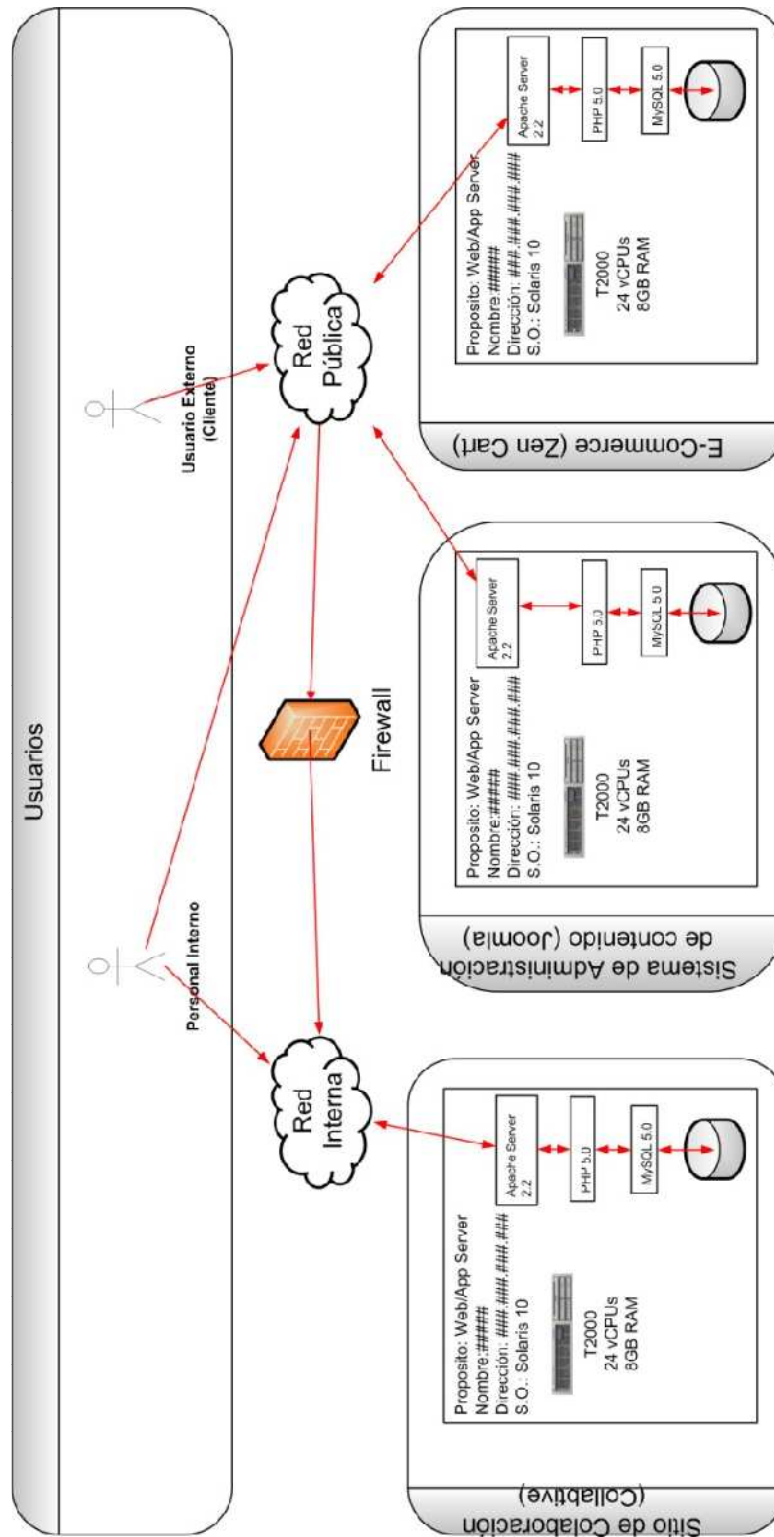


Figura 3.8 Mapa End-To-End de la Arquitectura Inicial

En el diagrama se puede observar que existen dos tipos de Usuarios, los usuarios que se conectarán a los equipos para su administración (Personal Interno) y los usuarios que accederán a los equipos para hacer uso de los servicios que estos ofrecen (cliente). Se puede apreciar que el Personal Interno puede acceder tanto a la “Red Interna” como a la “Red Pública” para llegar al servidor, esto es porque el Personal Interno debe de llevar a cabo la administración de los servidores y debe de contar con accesos a ambas redes. Por el contrario, el Cliente sólo puede acceder a los servicios que ofrece el servidor a través de la “Red Pública”, restringiendo de esta manera el acceso a los procesos más importantes de los servidores.

A través de la “Red Interna” y de la “Red Pública” puede accederse a los servidores, siendo más limitada la segunda conexión, además de que en la arquitectura se cuenta con un *Firewall* para restringir los accesos entre una y otra.

Se representan tres equipos de hardware Sun Fire T2000, cada uno con 8 GB en RAM y con 24 hilos de ejecución, representados en el diagrama como vCPUs (virtual CPUs). En el primer servidor, se tiene instalado y configurado el Sitio de Colaboración, en el segundo equipo se tiene el Sistema de Administración de Contenido y en el último, se encuentra el Comercio Electrónico.

### 3.2.3 Requerimientos y Características

Las características que tendrá el diseño para cumplir con los requerimientos de negocio en cuanto a la mejora del nivel de servicio son: disponibilidad del sistema, facilidad de recuperación en caso de falla, redundancia en servicios de red, facilidad de administración y como característica primordial en todo sistema, la seguridad.

- Disponibilidad del sistema: Con esta característica el sistema puede ofrecer el servicio de manera in-interrumpida por mayor tiempo.
- Facilidad de recuperación en caso de falla: Con esta característica el sistema puede volver a dar el servicio en menor tiempo, después de una falla.
- Redundancia en servicios de red: Este es un requerimiento común en los sistemas y servidores actuales, ya que en situaciones de mantenimiento y en situaciones de falla de componentes de comunicación externos al servidor, puede continuar con la operación normal.

- **Facilidad de administración:** Esta característica es necesaria en los sistemas actuales, debido a la complejidad que pueden llegar a alcanzar. En ocasiones administrar más de 100 instancias de sistema operativo resulta complicado si no se tienen las consideraciones necesarias.
- **Seguridad:** Dependiendo del tipo de información con que se cuente en el sistema, se tendrá mayor o menor grado de seguridad, ya sea por medio de mecanismos propios del servidor, el sistema operativo o mediante dispositivos más complejos.

### **3.2.4 Herramientas**

Para cumplir con los requerimientos antes descritos se hará uso de herramientas y dispositivos contemplados en el Capítulo 2, subtemas Herramientas de Hardware y Herramientas de Software, como se describe a continuación.

#### **Disponibilidad del sistema**

Lograr una buena disponibilidad en el sistema, por lo general, significa contar con dos elementos idénticos dentro del mismo servidor físico, en caso de falla de alguno de los elementos principales el segundo elemento es capaz de ofrecer la misma función, sin degradación o afectación en la calidad del servicio brindado. Las propiedades mencionadas en el capítulo 2 sobre Solaris 10 y del servidor Sun Fire T2000 permiten garantizar un buen nivel de disponibilidad para los Solaris Containers y las aplicaciones albergadas en ellos, con total transparencia y sin la necesidad de introducir un elemento adicional de software.

#### **Facilidad de recuperación en caso de falla**

Al presentarse un error en el sistema, sea este intencional o no, es común, que la integridad pueda verse comprometida.

Se puede llegar a contar con copias de seguridad o respaldos, suelen utilizarse hasta que se produce un incidente haciendo complicada la tarea de recuperación.

Al utilizar Solaris Containers es posible tomar una foto (snapshot) del sistema completo, con lo que en caso de que ocurra algún error o falla, se revierte a un estado anterior conocido del sistema de forma inmediata. Esta funcionalidad hace que la

recuperación del sistema se realice en un tiempo muy reducido, a comparación de los sistemas no virtualizados.

### **Redundancia en servicios de red**

Solaris proporciona IPMP (*IP Multipathing*), la cual es una técnica para la configuración de las tarjetas de red (*NIC, Network Interface Card*), que requiere de varios elementos para su funcionamiento: Vínculo IP, Tarjeta de Interfaz de red (NIC), Interfaz física y Grupo IPMP.

- **Vínculo IP** es el medio de comunicación, el cual puede pertenecer a una subred; también abarca redes Ethernet simples, Ethernet con puente, concentradores o redes ATM.
- **Interfaz física** es un dispositivo que da lugar a la comunicación entre un equipo y un vínculo IP, en este caso el servidor.
- **Interfaz de red** (NIC) es la placa de circuito física instalada en un equipo, la cual se utiliza en operaciones de comunicación con la red.
- **Grupo IPMP** es un conjunto de una o más interfaces físicas en el mismo sistema, este conjunto se conectan al mismo vínculo IPMP y pertenecen al mismo grupo.

Existen dos tipos de configuraciones, la del tipo **Activa-Activa** donde ambas interfaces se encuentran trabajando de manera normal, transmitiendo datos al igual que tráfico de sondeo, en contraparte la configuración **Activa-Reserva** como su nombre lo indica, consiste en que una de las interfaces se encuentra de reserva, es decir si se presenta una falla en la interfaz activa del grupo, la de reserva realiza todas las funciones de transmisión de datos, recibe el mismo tratamiento que las demás interfaces activas hasta que se repara la interfaz fallida, de esta forma se evitan tiempos muertos y se asegura la disponibilidad del sistema.

IPMP implementa detección de fallas en las interfaces del sistema mediante el tráfico de sondeos, el cual ayuda a determinar el estado de la interfaz y su NIC.

El sondeo se realiza en todas las interfaces que forman parte de un grupo IPMP, verifica que la conexión de la interfaz al vínculo IP este activa al igual que su ruta de transmisión o recepción, utiliza las direcciones de prueba para el intercambio de sondeos ICMP con otros destinos del vínculo IP, también comprueba la existencia de las interfaces

físicas al inicio del sistema y se encarga del “sensado” de la transmisión y recepción de paquetes en el puerto del conmutador. Si se llegara a presentar alguna de las fallas mencionadas, el “demonio” ***daemon in.mpathd*** identifica las fallas y a continuación implementa el procedimiento de conmutación para la recuperación tras la falla, de un modo transparente y automático. La conmutación por error consiste en relevar el acceso de red entre interfaces de un mismo grupo IPMP, de tal forma que se puedan migrar las funciones de una interfaz fallida a otra interfaz, dando lugar a que continúe de manera normal el tráfico de red.

Con la configuración *IP Multipathing* (IPMP) del tipo **Activa-Reserva** en las dos tarjetas de red del servidor Sun Fire T2000 se proporciona redundancia y recuperación ante fallas a los recursos de red físicos que utilizarán los Solaris Containers (figura 3.9), incrementando de manera significativa la disponibilidad de la red, ya que si alguno de los adaptadores de red pertenecientes al grupo de IPMP falla, todos los servicios de red recurren al otro adaptador del grupo manteniendo la conectividad e impidiendo la interrupción de las conexiones ya establecidas.

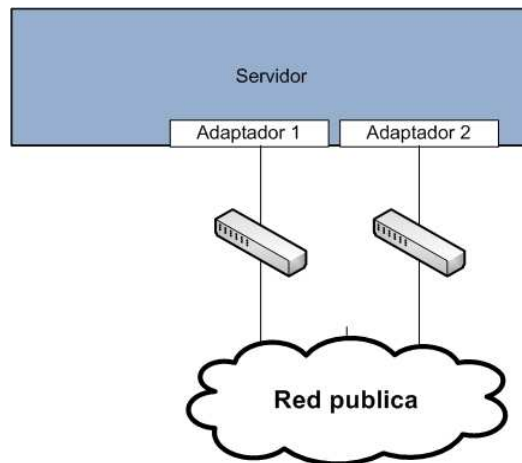


Figura 3.9 Diagrama de IPMP

### Facilidad de administración

Al contar con sistema operativo host albergando los Solaris Containers, la administración y mantenimiento se simplifica, realizando los cambios necesarios de configuración de los recursos asignados para tener un mejor rendimiento conforme lo vaya demandando cada una de las aplicaciones. Además de reducir el número de



servidores físicos a administrar, lo que conlleva tiempo y esfuerzo para las personas encargadas, en especial cuando se cuenta con un gran número de servidores; cada Solaris Container se apaga, reinicia y enciende desde el sistema operativo host, teniendo la posibilidad de realizar estas actividades remotamente sin tener que estar de frente al servidor físico. En caso de requerir un mantenimiento mayor en el sistema operativo host, el servidor Sun Fire T2000 cuenta con un puerto serial de administración remota (*system controller*), el cual cuenta con una dirección IP y un usuario de administración general, capaz de realizar las tareas antes mencionadas del sistema operativo host.

### **Seguridad**

Una empresa puede tener una gran variedad de servicios web, ya sean para procesos internos, a los que sólo tendrán acceso los empleados y procesos externos necesarios para la comunicación con sus clientes. Aunque los servicios web pueden tener elementos de funcionamiento en común, es recomendable que cada servicio esté aislado de los demás para prevenir problemas de seguridad y prevenir que los errores que puedan surgir no afecten al resto de las aplicaciones. Para prevenir esta situación cada Solaris Container contará con su propio nombre de sistema (*hostname*), dirección IP, sistemas de archivos propios, usuario administrador (*root*), etc. Si un error se presentara en un Solaris Container, los servicios que éste provee serían los únicos afectados, si el error provoca la caída del servicio, este puede ser encendido, reiniciado o apagado sin afectar el resto de los demás servicios y Solaris Containers existentes. Las aplicaciones estarán protegidas de errores y de problemas de seguridad que puedan afectar su disponibilidad o su desempeño.

Para poder cumplir con las características aquí descritas, los requerimientos que deben contemplarse son:

1. Mínimo, 2 tarjetas de Red Ethernet con al menos 2 puertos, esto para cumplir con redundancia en red, poder configurar IPMP y además, para que cada Servidor Virtualizado cuente con una dirección de internet dedicada.
2. Dos fuentes de poder redundantes en el equipo de hardware.
3. Si se cuentan con servicios que necesitan realizar respaldos de datos a una red SAN (Storage Area Network), entonces el equipo de hardware debe de contar con al menos, dos adaptadores de red de fibra (Host Bus Adapter, HBA) para contar con alta disponibilidad en respaldos.

4. Puerto de administración (RJ45 ó puerto serial), así como también debe contemplarse una dirección IP para la administración del servidor.

### 3.2.5 Diseño y Arquitectura Propuesta

Un escenario común que suele presentarse en las empresas, es la prestación de servicios hacia sus clientes a través de Internet, ya sea para ventas o para la contratación de servicios. Al ser el núcleo del negocio, requieren tener a su disposición un cierto número de recursos disponibles, en caso de un aumento en la demanda. Esta aplicación generalmente es conocida como Comercio Electrónico (*e-commerce*), conformada por diferentes tipos de servidores: servidor de Base de Datos, servidor de Aplicación y servidor Web y todos interactúan entre sí. Esto conllevaba a las empresas verse en la necesidad de comprar un servidor para cada una de las aplicaciones y al presentarse un aumento en la demanda, tal vez no se contaba con los recursos para soportarla, provocando degradación en el servicio y pérdidas monetarias a la empresa.

Para evitar la subutilización de recursos de hardware, así como el tiempo ocioso de los sistemas, se integran en el mismo servidor otras aplicaciones necesarias para la operación de la empresa, como lo es el manejador de contenido y los sitios de colaboración. Estas aplicaciones funcionan con un servidor de base de datos, un servidor de aplicación y un servidor web, al no ser aplicaciones tan concurrentes no demandan tantos recursos, como lo haría la aplicación de Comercio Electrónico, ya que suelen ser accedidas solo por personal de la empresa y no son el núcleo de negocio.

Estas aplicaciones serán virtualizadas a través de Solaris Containers sobre un mismo servidor físico, existirá un contenedor para “Servidor de Aplicación y Web” y un segundo contenedor para el “Servidor de Base de Datos”, teniendo dos Solaris Containers para el funcionamiento de cada aplicación.

Este diseño considera la división en 2 capas, que a diferencia de modelo de 3 capas, consolida al servidor Web con el servidor de Aplicaciones dejando al servidor de Base de Datos independiente, esto es debido al software que se está utilizando, Apache (Web) y PHP (Aplicaciones) deben trabajar en la misma instancia de Sistema Operativo, ya que PHP trabaja como un módulo de Apache.

El diseño además contempló cada una de los contenedores listados a continuación, en primer lugar para cubrir el requerimiento de facilidad de administración, ya que al tener la base de datos separada del servidor web/aplicación, en una ventana de mantenimiento es

mucho más sencillo realizar las acciones necesarias como dar de baja los servicios o aplicar parches. En segundo lugar esta división también previene que los recursos que estén reservados para la base de datos sean utilizados por el servidor web/aplicación y viceversa, de esta forma el rendimiento es más estable, ya que un servidor web y aplicaciones usa mayor cantidad de CPUs debido a que atiende mayor cantidad de usuarios, reflejados en conexiones simultaneas, y en el caso de las aplicaciones aquí utilizadas, la base de datos requiere menor cantidad de CPUs, ya que realizan consultas sencillas, es decir, no realizan mucho procesamiento en la información, en su mayoría son consultas.

Los seis Solaris Containers, con la aplicación albergada son:

- Servidor de base de datos para Comercio Electrónico
- Servidor de aplicación y web para Comercio Electrónico
- Servidor de base de datos para Manejador de Contenido
- Servidor de aplicación y web para Manejador de Contenido
- Servidor de base de datos para Sitios de Colaboración
- Servidor de aplicación y web para Sitios Colaboración

Con los recursos asignados como se indican en la tabla 3.5 a continuación:

Aplicación	Componente de Software	Cantidad de VCPUs	Tope de Memoria
Comercio Electrónico	Base de Datos	4 VCPUs	1 GB
Comercio Electrónico	Web/Aplicaciones	8 VCPUs	1 GB
Manejador de Contenido	Base de Datos	2 VCPUs	1 GB
Manejador de Contenido	Web/Aplicaciones	4 VCPUs	1 GB
Sitio de Colaboración	Base de Datos	2 VCPUs	1 GB
Sitio de Colaboración	Web/Aplicaciones	2 VCPUs	1 GB
Zona Global (Administración)		2 VCPUs	1 GB

De esta forma los Contenedores para la aplicación de Comercio Electrónico tienen más recursos asignados, puesto que es la aplicación que más demanda esperaríamos, además de que en ella se llevan a cabo transacciones. El Contenedor de Manejador de Contenidos, requiere menor cantidad de recursos ya que no realiza transacciones, solamente

consultas por parte de los usuarios y actualizaciones por parte de los administradores. La aplicación de Sitio de Colaboración, al ser un sitio para usuarios internos, tiene menor demanda, por ello requiere menor cantidad de recursos. Por último, la Zona Global es la que administra a todos los Contenedores, pero requiere recursos independientes para poder operar.

El diseño hecho para la virtualización de las aplicaciones de una empresa se muestra en la figura 3.10. La configuración de cada Solaris Container constará de su propio nombre de equipo, dirección ip, usuarios y la asignación de recursos será dinámica.

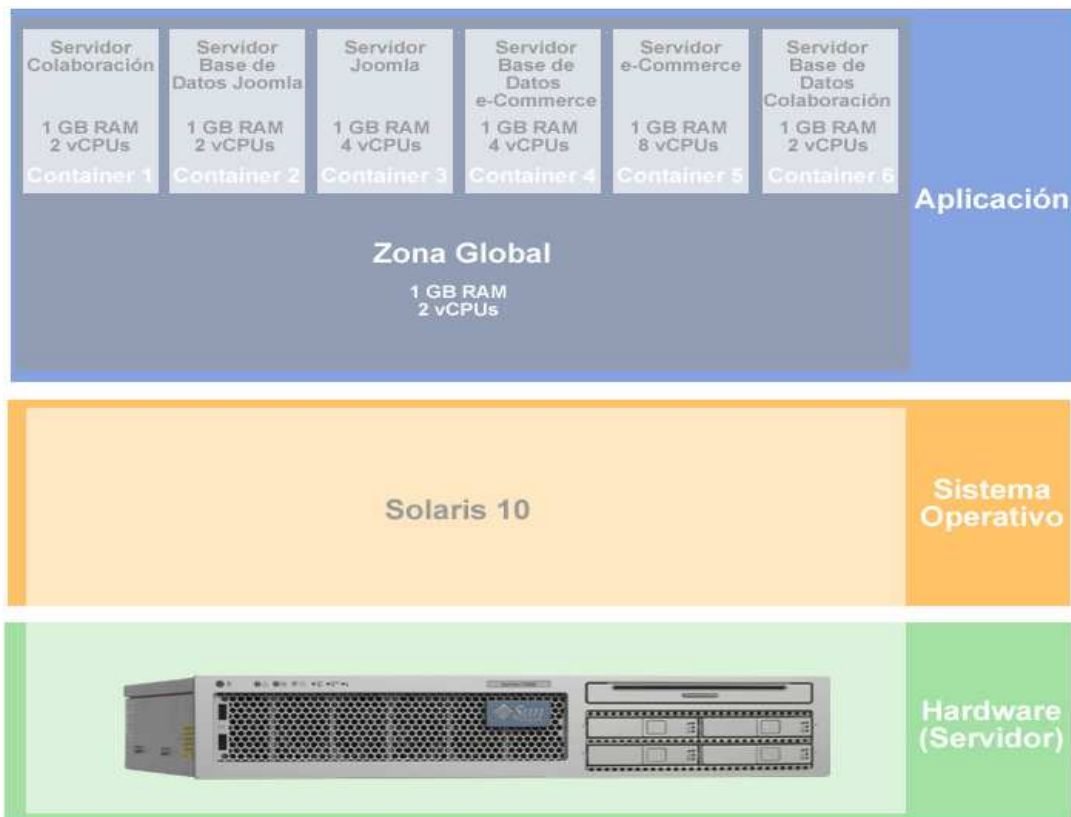


Figura 3.10 Arquitectura Propuesta y Recursos asignados

A continuación (figura 3.11) se muestra el diagrama End-To-End de la Solución de Virtualización para la Arquitectura propuesta.

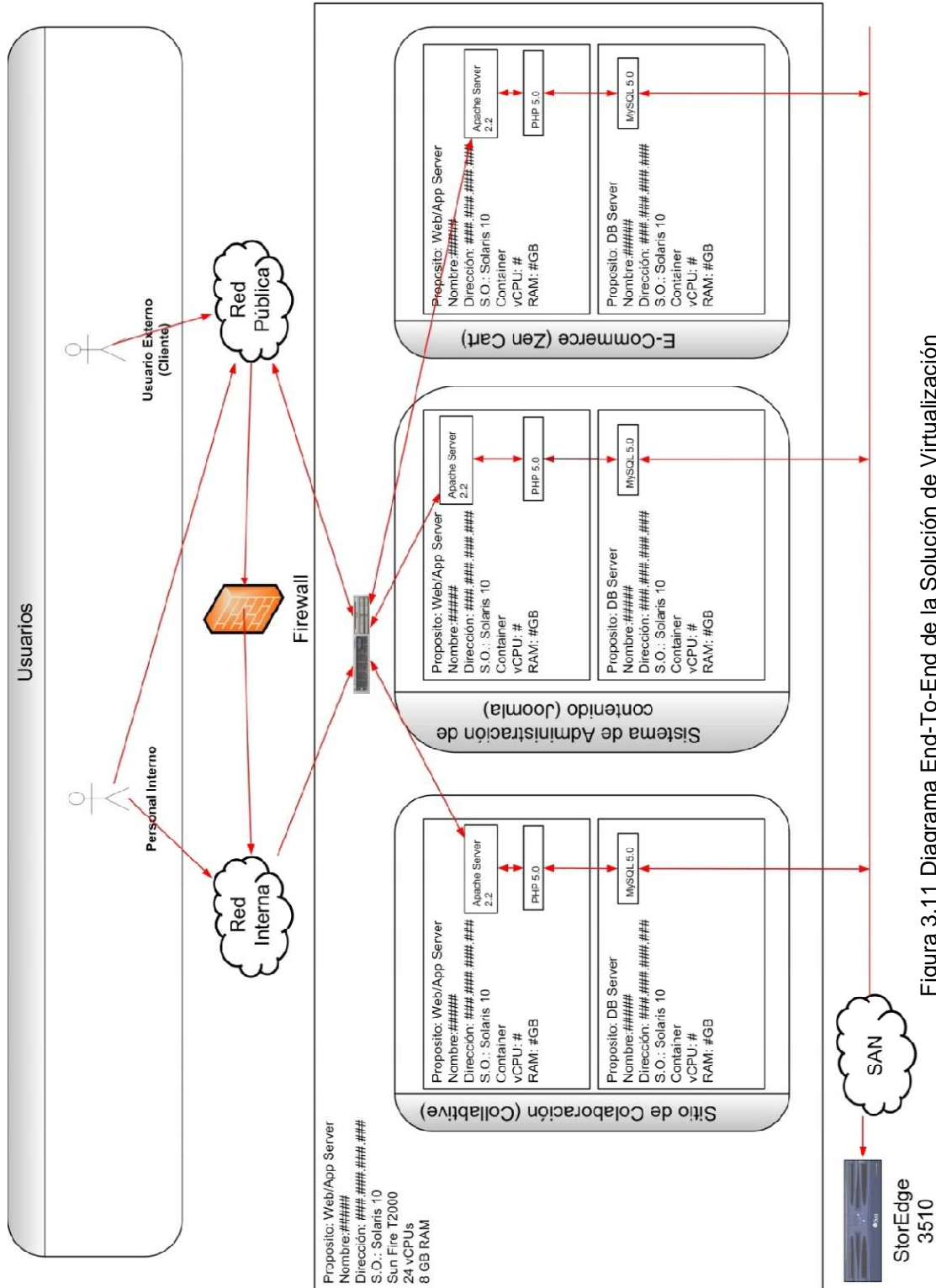


Figura 3.11 Diagrama End-To-End de la Solución de Virtualización

En el diagrama podemos observar una capa superior llamada “Usuarios”, esto representa los usuarios que se conectarán al equipo. Los usuarios pueden ser personal interno de la misma compañía, que necesitan conectarse al servidor para llevar a cabo actividades de administración del equipo y sus servicios. También pueden conectarse al servidor, usuarios externos que quieran acceder a un servicio brindado por el servidor. Por esto mismo, se cuenta con una “Red Interna” para los usuarios administrativos y se cuenta con una “Red Pública”, para los clientes que deseen utilizar los servicios instalados en el equipo.

Contar con una “Red Interna”, “Red Pública” y con un Firewall es altamente recomendable y necesario, ya que esto limita el acceso de los usuarios al servidor y provee un nivel de seguridad al equipo y los servicios que proporciona. También debe de considerarse asignar los privilegios necesarios a cada perfil diferente de usuario, es decir, el personal interno que deba administrar el equipo debe de contar con ciertos privilegios para que pueda administrar el servidor y por el contrario, los usuarios externos no deben de contar con los mismos permisos que un administrador, ya que éstos sólo podrán acceder a los servicios pero sin efectuar cambios en ellos.

En la arquitectura se puede observar que sobre el servidor Sun Fire T2000 está instalado el Sistema Operativo Solaris 10 y sobre éste, habitan los servicios virtualizados: el Sitio de Colaboración, el Sistema de Administración de Contenido y el Comercio Electrónico; cada uno con sus respectivos recursos asignados, con su propia Base de Datos y con su Servidor de Aplicación.

En la parte baja del diagrama se representa un equipo de almacenamiento Sun StorEdge 3510, en el cual cada Base de Datos de los diferentes servicios, cuenta con un espacio en Disco asignado. Las Bases de Datos se conectan a este disco a través de la red SAN (*Storage Area Network*).

Para las etapas de Implementación y de Evaluación contempladas en la Metodología General, está dedicado el Tema 4 y el Tema 5, respectivamente, para el desarrollo y explicación de estos temas.

# *Capítulo 4*

---

*Este capítulo se centra en la implementación, se detallan y explican los trabajos necesarios para la instalación y configuración de Solaris Containers.*

## **4. IMPLEMENTACIÓN**

Siguiendo la metodología que se ha descrito en el capítulo anterior, se decidió incluir en capítulos separados los temas de Implementación y Evaluación, debido a la complejidad y a la cantidad de información que se presenta en cada una de ellas. El presente capítulo abarca la etapa posterior al diseño desde la adquisición de los equipos, la adecuación del espacio, hasta la liberación de los servicios, de acuerdo a la infraestructura con que se cuente. Para el caso de este trabajo, la parte de adquisición y montado físico del equipo no aplica, sin embargo en la gran mayoría de los proyectos es necesario considerarlo.

El capítulo está dividido en diferentes subtemas de acuerdo con la configuración específica que llevaremos, los cuales son:

- Instalación del Sistema Operativo base
- Configuración del Sistema Operativo base
  - Configuración de Red
  - Configuración de Hardening
  - Configuración de Espejo de Discos (RAID 1)
- Configuración de Solaris Containers
  - Configuración del Pool de Recursos para las Zonas
  - Definición de Zonas
- Instalación de Webstack
  - Servidor de Aplicaciones
  - Servidor de Base de Datos
- Configuración de Webstack

Estos subtemas abarcan la configuración llevada a cabo, de tal forma que sea posible entender progresivamente cada uno de los puntos de implementación.



Figura 4.1 Servidor T2000

#### 4.1 Instalación del Sistema Operativo base

El sistema operativo Solaris 10 puede ser instalado mediante la red o desde un medio (cd/dvd). Los requerimientos mínimos para su instalación son:

- Memoria Ram mínima de 256 MB
- Espacio en disco duro 5GB
- Procesador 200 MHz o superior

A continuación se resume la instalación de Solaris 10, cabe mencionar que el proceso detallado de la instalación del sistema operativo base se encuentra en el Anexo 2.



El Programa de instalación de Solaris ofrece una opción mediante interfaz gráfica de usuario (GUI) y otra mediante texto, ya sea desde una conexión al puerto serial o desde un monitor conectado al servidor. De igual forma, se pueden utilizar otras opciones de Instalación como son el Jumpstart y el Flash Archive. En la primera podemos definir las características del sistema a instalar a partir de otro sistema existente; y el método de instalación Flash Archive consiste en instalar y configurar un sistema base del que se realiza un respaldo Flash Archive, éste respaldo se usa para instalar el sistema operativo en uno o varios sistemas más.

Para el caso del servidor Sun Fire T2000 la instalación inicia en modo de texto, viendo la salida por medio del puerto serial, ya que no se cuenta con interfaz grafica. La primera pantalla es la de configuración del sistema y requiere que se elija el idioma del teclado y de la Interfaz, seleccionamos **Inglés**.

En las pantallas siguientes proporcionamos el nombre del servidor el cual debe ser único, en este caso se le asignó el nombre **UXSOP001** así como algunos de los parámetros de red: indicamos que pertenece a una subred, así como su máscara de subred **255.255.255.0**, establecemos que se trata de una asignación de IP manual y no por **Dynamic Host Configuration Protocol (DHCP)**, proporcionamos la dirección IP la cual también debe ser única **192.9.198.190**, deshabilitamos la opción de *Ipv6* ya que no vamos a utilizar ese protocolo, posteriormente especificamos la dirección IP del Router que vamos a utilizar, **192.9.198.1**. (Figura 4.2).

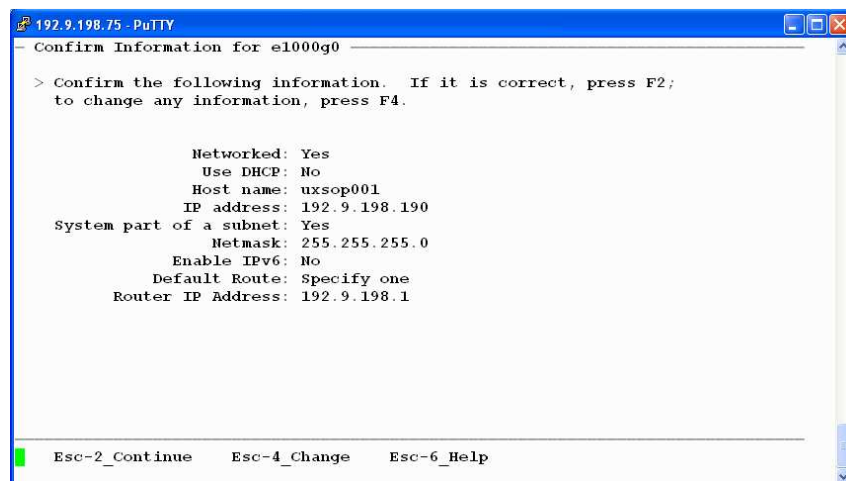


Figura 4.2 Instalación del Servidor

A continuación configuramos las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad*

*estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros. Es por esta razón que elegimos la Seguridad Estándar de UNIX. El siguiente paso es indicar si el servidor tendrá o no un servicio denominado **Name Service**, ya sea NIS, DNS o LDAP en nuestra configuración seleccionamos **NONE**, ya que el servidor no tendrá ninguno de los servicios antes mencionados y posteriormente seleccionamos **Use the NFSv4 domain derived by the system** con lo cual se genera un nombre de dominio de forma automática el cual se utiliza para la resolución interna. La siguiente sección requiere la selección de información relacionada con la ubicación, en nuestro caso en Time Zone es **Americas**, en Country **Mexico y Central Time** para la ciudad de México (Figura 4.3).

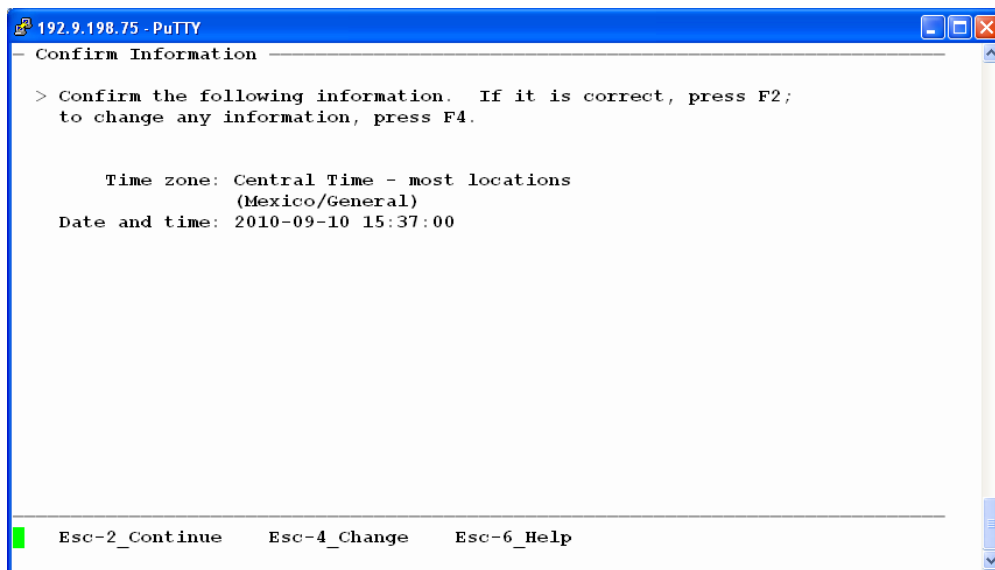


Figura 4.3 Instalación del Servidor

El usuario root es el que tiene el mayor nivel de privilegios y es por esto que el instalador solicita el **password de root**. Finalmente establecemos la configuración para **Remote Services Enabled**, que son todos los servicios de red para clientes remotos, por razones de seguridad seleccionamos la opción de **NO**, donde deshabilitamos estos servicios con la excepción de Secure Socket Layer (SSH), esto se realiza con la finalidad de reducir la superficie de ataque mediante la desactivación de los servicios de red que no se van a utilizar.

De ser necesario se configuran los servicios de forma independiente después de la finalización de la instalación, una vez hecho esto se muestra el progreso de la instalación **System Identificación Complete**.

Una vez que el sistema inicializa procedemos a seleccionar el sistema de archivos que se va a utilizar, se selecciona **UFS** que es el nativo en Solaris para la instalación del Sistema Operativo y el **ZFS** se utilizará en las zonas ya que ofrece un mayor espacio para los archivos, mejora enormemente la administración y cuenta con un buen nivel de seguridad de datos, a continuación preferimos la opción de **Entire Distribution plus OEM support** del menú de software, esta opción incluye controladores de hardware adicionales incluso para hardware que se va incorporar posteriormente a la instalación, además de los paquetes para el grupo de software de distribución completa, a continuación seleccionamos el disco duro para la instalación del software **c1t0d0**. En el paso siguiente, configuramos el sistema de archivos, se ofrecen las opciones de **Auto Layout**, esta opción genera una sola partición de disco de arranque completo, mientras que la opción **Manual Layout** permite asignar los recursos necesarios para cada partición; preferimos la segunda opción ya que es más granular y con esto se distribuyen mejor los recursos, lo que da lugar a un mejor y mayor eficiencia.

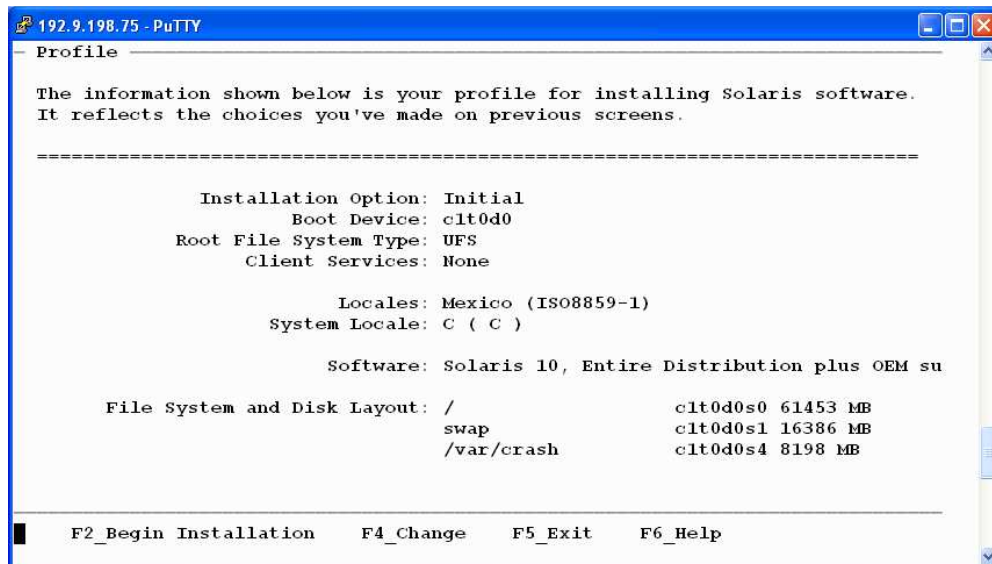


Figura 4.4 Configuración del disco

Una vez hecho esto comienza la instalación donde se nos informan los cambios que se realizaron en disco duro (Figura 4.4).

Posteriormente, el instalador muestra el número de bytes instalados al igual que los faltantes, así como una breve descripción de la herramienta que se encuentra instalando y en la parte inferior aparece el porcentaje de instalación en una escala de 0 a 100, donde 100 % es la finalización del proceso de instalación Figura (4.5).

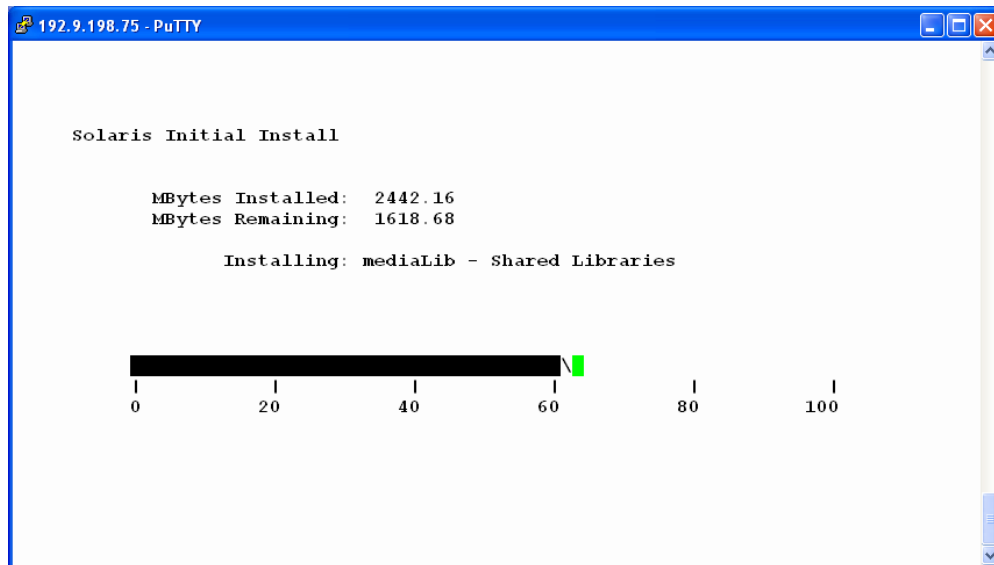


Figura 4.5 Proceso de Instalación

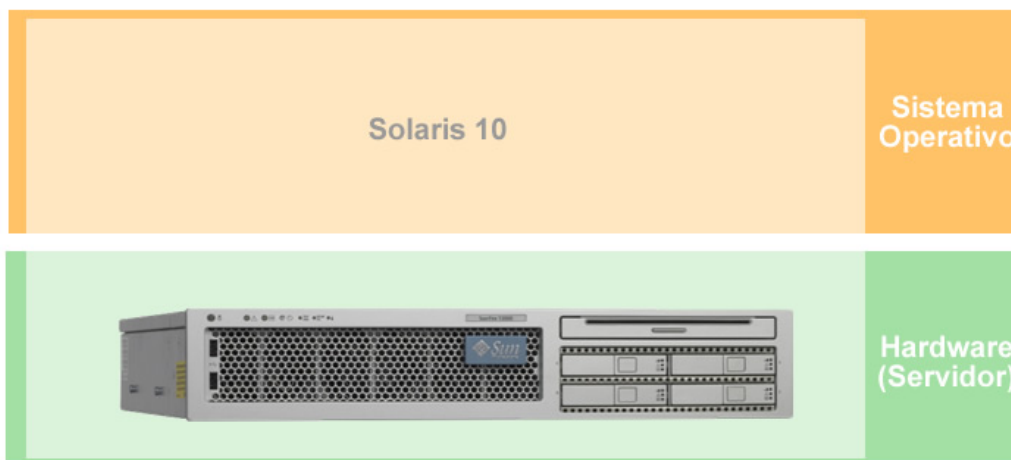


Figura 4.6 Sistema Operativo Base

## 4.2. Configuración del Sistema Operativo base

Posterior a la instalación del Sistema Operativo, en donde se instalan los archivos base y configuraciones mínimas que permiten operar el servidor, es necesario realizar configuraciones con las que se cumplan los requerimientos de diseño.

De acuerdo al diseño establecido para nuestro trabajo de Tesis, la configuración del Sistema Operativo base incluye los siguientes rubros, los cuales se detallan más adelante en este subtema:

- Configuración de Red
- Configuración de Hardening
- Configuración de Espejo de Discos

4.2.1. Red

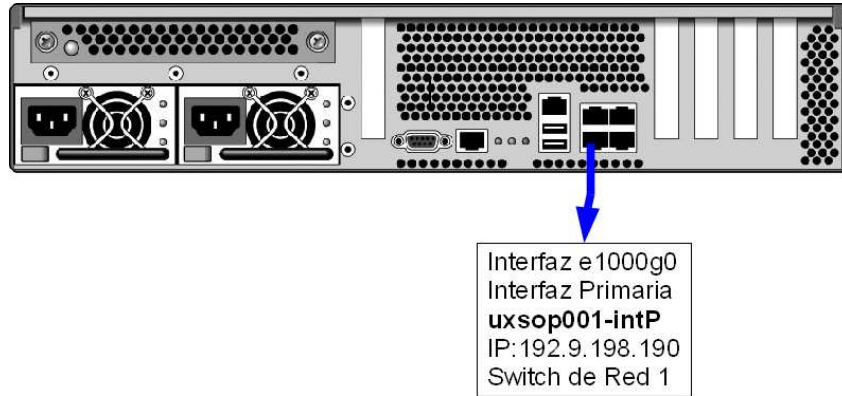


Figura 4.7 Configuración Inicial de Red

Las interfaces de red deben cumplir con ciertos requisitos para poder configurarse utilizando IPMP, deben tener direcciones MAC únicas es por ello que en caso de tener una dirección MAC compartida como sucede en equipos SPARC es importante cambiar su configuración, de tal forma que cada interfaz tenga su propia MAC, adicionalmente deben utilizar el mismo medio y compartir el mismo vínculo IP.

A continuación se describe brevemente la configuración IPMP del tipo **Activa-Reserva** de las interfaces de red del servidor Sun Fire T2000.

En el servidor UXSOP001 mediante el uso del comando **cat**, listamos el contenido del archivo `/etc/hosts`, verificamos que este archivo contenga IP, nombre *FQDN* (*Fully Qualified Domain Name*) del servidor. Después comprobamos las interfaces disponibles, con el parámetro **show-dev** del comando **dladm**, se listan todos lo interfaces disponibles y su estado. Observamos que se tienen 2 interfaces de red, **e1000g0** y **e1000g2**, respectivamente tienen una velocidad de 100Mbps y están conectadas al mismo vínculo IP.

```

uxsop001 / # dladm show-dev
e1000g0      link: up      speed: 100  Mbps      duplex: full
e1000g1      link: unknown speed: 0    Mbps      duplex: half
e1000g2      link: up      speed: 100  Mbps      duplex: full
e1000g3      link: unknown speed: 0    Mbps      duplex: half
    
```

La interfaz e1000g2 no se encuentra en línea, es por ello que utilizamos el comando **ifconfig** con el parámetro **plumb**, lo que pone en línea la interfaz y con el parámetro **-a** verificamos el estatus de la interfaz, una vez hecho esto establecemos los parámetros de red de la interfaz: dirección IP y su máscara de subred.

```

uxsop001 / # ifconfig e1000g2 plumb
uxsop001 / # ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232
index 1
    inet 127.0.0.1 netmask ff000000
e1000g0: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.9.198.190 netmask ffffffff broadcast 192.9.198.255
    ether 0:3:ba:d8:fb:ce
e1000g2: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
    inet 0.0.0.0 netmask 0
    ether 0:3:ba:d8:fb:d0
uxsop001 / # ifconfig e1000g2 192.9.198.130 netmask 255.255.255.0 up
    
```

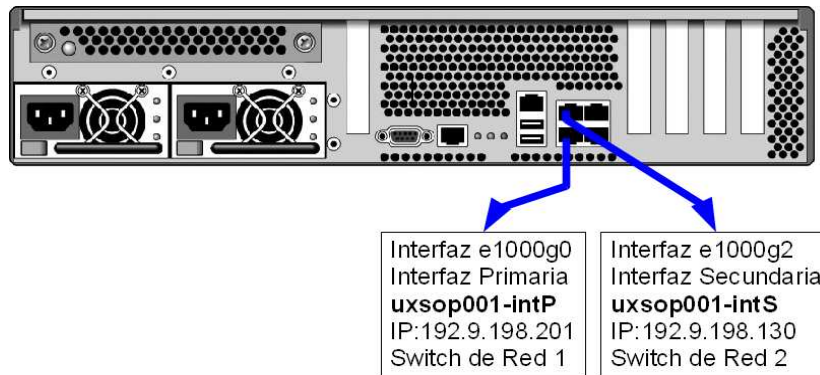


Figura 4.8 Configuración de las interfaces de red

La configuración IPMP del tipo Activa-Reserva requiere que una interfaz sea la primaria la cual llamaremos **uxsop001-intP** y la otra de reserva **uxsop001-intS**, es por esto que establecemos estos parámetros en el archivo **/etc/hosts**.

```

uxsop001 / # cat /etc/hosts
#
# Internet host table
#
::1    localhost
127.0.0.1    localhost
192.9.198.190    uxsop001          uxsop001.tesis.com    loghost

#Direccion IP Interfaz principal
192.9.198.201    uxsop001-intP

#Direccion IP Interfaz secundaria
192.9.198.130    uxsop001-intS
    
```

La interfaz principal **uxsop001-intP** y secundaria **uxsop001-intS** las configuramos para que pertenezcan al mismo grupo, el grupo tiene un nombre para su distinción y en nuestro

caso le llamaremos **ipmp0**, posteriormente se establece una dirección de prueba, esto con la finalidad de que realice el proceso de la detección de fallos.

Este proceso se basa en sondeos en una interfaz específica mediante los parámetros -**failover deprecated up**, todo esto se realiza en el directorio *etc* en los archivos *hostname.e1000g0* y *hostname.e1000g2*, ya que de lo contrario se perderá esta configuración al reinicio del equipo. Únicamente en la interfaz principal definimos la interfaz lógica mediante **addif**.

```

uxsop001 / # cat /etc/hostname.e1000g0
uxsop001-intP netmask + broadcast + deprecated -failover group ipmp0 up \
addif uxsop001 netmask + broadcast + up

uxsop001 / # cat /etc/hostname.e1000g2
uxsop001-intS netmask + broadcast + deprecated -failover group ipmp0 up
    
```

Finalmente reiniciamos el servidor que ya tiene configurado IPMP, una vez que el equipo ha iniciado verificamos la correcta configuración de las interfaces, con lo que se garantiza la redundancia en la red (Figura 4.9).

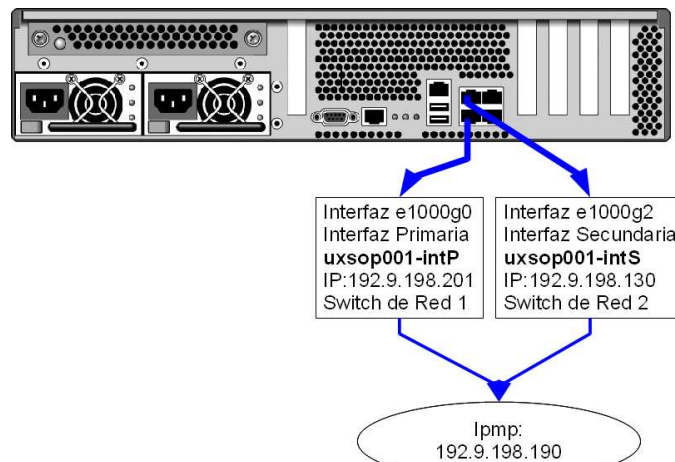


Figura 4.9 Configuración IPMP

#### 4.2.2. Hardening

Después de la instalación Solaris 10, se deben realizar ciertas tareas de configuración y personalización del sistema operativo, al ser éste el host base para los Solaris Containers, es necesario preparar el sistema para prevenir ataques de seguridad.

El objetivo es remover servicios vulnerables o innecesarios, eliminar los huecos de seguridad que puedan existir y mejorar la configuración de los servicios ofrecidos para

estar preparados ante amenazas que puedan atentar contra la aplicación ofrecida, así como resguardar la información que se maneja en el servidor y en los Solaris Containers.

### Seguridad física

El servidor Sun Fire T2000 cuenta con una memoria RAM no volátil (NVRAM) denominada OpenBoot PROM (OBP), cuenta con una interfaz de software que se encarga de verificar el estado del hardware para después arrancar el sistema operativo, ayudar al diagnóstico de problemas y obtener información sobre el servidor, todas estas operaciones son por medio de línea de comandos.

Es posible acceder a la OBP por medio de la controladora del sistema (system controller) mandando la sentencia de **break**, o bien terminando la ejecución del sistema operativo. Esto representa un problema de seguridad ya que alguien con acceso al servidor conectado a la controladora del sistema, podría enviar la sentencia de **break** intencionalmente o por error, lo que ocasionaría la detención de todos los procesos que se encuentren ejecutándose en el sistema operativo, además de proporcionar total control sobre el sistema y sus parámetros de configuración a nivel de hardware.

Para evitar esta situación deshabilitamos la función de **break**. Se coloca en el archivo de configuración `/etc/system` la directiva: **KEYBOARD\_ABORT = disable**. Evitando de esta manera la detención del sistema operativo enviando la función `break`.

A nivel de OBP se configuro el parámetro `security-mode`, en modo “**command-secure**”, para que al reiniciar el servidor, y se intente el arranque con un dispositivo que no sea el indicado en la configuración inicial solicite la contraseña proporcionada.

### Optimización de parámetros de red

Los parámetros de red que se cargan al momento en que inicia Solaris 10, necesitan ser modificados para optimizar las funciones de red. Estos parámetros por default presentan los siguientes valores:



```
ip_respond_to_echo_broadcast 1
ip_forward_directed_broadcasts 0
ip_strict_dst_multihoming 0
ip_ignore_redirect 0
ip_forwarding 0
ip_forward_src_routed 0
ip_ire_arp_interval 1200000
arp_cleanup_interval 300000
ip_respond_to_timestamp 0
ip_respond_to_timestamp_broadcast 0
```

Al modificar estos parámetros se pueden prevenir ataques y errores, por ejemplo: errores por un ping de broadcast, se pueden bloquear los paquetes de broadcast hacia la red, prevenir **Spoofing**, evitar un redireccionamiento de IP, eliminar paquetes de enrutamiento, se puede redefinir el tiempo de expiración de las tablas de ARP y broadcast para la sintonización de reloj y fecha ICMP, entre otros.

Para modificar los valores se crea un script de inicio en la ruta */etc/rc2.d/S99netconfig*.

```
#!/bin/ksh
/usr/sbin/ndd -set /dev/ip ip_respond_to_echo_broadcast 0
/usr/sbin/ndd -set /dev/ip ip_forward_directed_broadcasts 0
/usr/sbin/ndd -set /dev/ip ip_strict_dst_multihoming 1
/usr/sbin/ndd -set /dev/ip ip_ignore_redirect 1
/usr/sbin/ndd -set /dev/ip ip_forwarding 0
/usr/sbin/ndd -set /dev/ip ip_forward_src_routed 0
/usr/sbin/ndd -set /dev/ip ip_ire_arp_interval 60000
/usr/sbin/ndd -set /dev/arp arp_cleanup_interval 60000
/usr/sbin/ndd -set /dev/ip ip_respond_to_timestamp 0
/usr/sbin/ndd -set /dev/ip ip_respond_to_timestamp_broadcast 0
```

## Configuración de valores del kernel

Para optimizar el nivel de seguridad del Sistema Operativo Solaris, existen diversos ajustes que se pueden llevar a cabo a nivel de kernel. En Solaris, existe el archivo */etc/system* que contiene parámetros específicos del kernel.

Algunos de los ajustes que deben de considerarse para mantener un buen nivel de seguridad, son por ejemplo:

- Lista de Ejecutables

Algunos programas que atacan las vulnerabilidades de seguridad de los sistemas (también conocidos como “*exploits*”), toman ventaja de la lista de ejecutables de kernel del

sistema para atacarlo. Estos programas intentan sobrescribir partes especiales de un ejecutable privilegiado de kernel (es decir, un programa de kernel que cuente con permisos para controlar el sistema) para poder controlarlo. En Solaris, estos “*exploits*” pueden evadirse haciendo de la lista de ejecutables de kernel, una lista *no ejecutable*. Agregando las siguientes líneas al archivo `/etc/system` se puede evitar una vulnerabilidad del sistema:

```
set noexec_user_stack=1
set noexec_user_stack_log=1
```

Con estas variables definidas el sistema reportará en un archivo log cualquier intento de ejecución no permitido así como el nombre de usuario que intento realizar la ejecución. Esto permitirá que Solaris cuente con más resistencia ante este tipo de ataques.

- Tamaño de los archivos Core

Este tipo de archivos se encargan de guardar información sobre programas o procesos que durante su ejecución terminaron de forma inesperada, por lo que se utilizan para la detección de errores, debido al gran numero de información que pueden contener es necesario limitar el tamaño de estos ya que pueden en un momento dado ocupar demasiado espacio en disco. Esto lo hacemos con el parámetro.

```
set sys:coredumpsize = 0
```

### **Deshabilitar servicios innecesarios**

Los servicios en un sistema operativo Solaris 10 son ofrecidos por el demonio **inetd** encargado de manejar las conexiones que llegan al servidor, cuando se realiza una petición a un servicio de red, el demonio **inetd** interpreta la solicitud y lo envía al servicio encargado de esa solicitud. En Solaris 10 el archivo que contiene la descripción de los servicios de red es `/etc/inetd.conf` y el archivo con el número de puerto que utilizan los servicios es `/etc/services`. Si estos servicios de red no son imprescindibles para el funcionamiento del sistema, se puede prescindir de ellos comentando con un signo “#” su declaración en los archivos descritos, por lo que al recibir una petición de un servicio que nosotros no hemos habilitado el servidor no responde a la petición por no encontrar su descripción.

Otros servicios de red son ofrecidos por demonios independientes que se ejecutan por medio de scripts declarados en los directorios de los diferentes niveles de ejecución del sistema al momento en que el servidor inicia.

Los directorios de niveles de ejecución del sistema (run level) contienen los scripts con los procedimientos para inicializar o detener los demonios de los servicios de red a inicializar cuando el sistema cambia de estado a alguno de los niveles de ejecución.

Los directorios y el Nivel les de ejecución de Solaris 10, se muestran en la tabla 4.1:

Directorio	Nivel de ejecución
/etc/rcS.d	single user
/etc/rc0.d	Shutdown
/etc/rc1.d	Start
/etc/rc2.d	multi-user
/etc/rc3.d	multi-user (por default)
/etc/rc4.d	multi-user (sin utilizar)
/etc/rc5.d	shutdown y poweroff
/etc/rc6.d	shutdown y reboot

Para evitar que se ejecuten los scripts al momento de iniciar el sistema, se detiene la ejecución del demonio con el argumento **“stop”** y en los directorios */etc/rc2.d* y */etc/rc3.d* se renombran los scripts, que son identificados con la letra “S” seguido de un número y el nombre del servicio.

```

/etc/rc3.d/S16boot.server stop
mv /etc/rc3.d/S16boot.server /etc/rc3.d/NO_S16boot.server

/etc/rc3.d/S50apache stop
mv /etc/rc3.d/S50apache /etc/rc3.d/NO_S50apache

/etc/rc2.d/S99dtlogin stop
mv /etc/rc2.d/S99dtlogin /etc/rc2.d/NO_S99dtlogin

/etc/rc3.d/S90samba stop
mv /etc/rc3.d/S90samba /etc/rc3.d/NO_S90samba

/etc/rc3.d/S80mipagent stop
mv /etc/rc3.d/S80mipagent /etc/rc3.d/NO_S80mipagent

/etc/rc3.d/S52img stop
mv /etc/rc3.d/S52img /etc/rc3.d/NO_S52img

/etc/rc3.d/S16boot.server stop
mv /etc/rc2.d/S70uucp /etc/rc2.d/NO_S70uucp

/etc/rc2.d/S47pppd stop
mv /etc/rc2.d/S47pppd /etc/rc2.d/NO_S47pppd

```

Posteriormente en caso de requerir la ejecución de alguno de estos servicios bastaría con pasarle al script renombrado el argumento **“start”**.

### Actualización del Sistema Operativo

Las actualizaciones de Solaris 10 son necesarias para corregir problemas, errores o debilidades en seguridad del Sistema Operativo. Se proveen por medio de parches, un parche es un conjunto de archivos y directorios que actualizan o corrigen archivos y directorios existentes pero que se ha detectado que puedan presentar errores o ser susceptibles a ataques de seguridad.

Los parches que son necesarios para la correcta ejecución del sistema son recomendados y los de seguridad para prevenir los ataques o vulnerabilidades. Estos son distribuidos al público en general por Internet, sin embargo, existen parches para errores específicos los cuales solo están disponibles para clientes con un contrato de mantenimiento.

Una vez descargados el conjunto de parches recomendados y el conjunto de parches de seguridad se transfieren a la carpeta `/var/tmp` del servidor para su instalación (Figura 4.10).

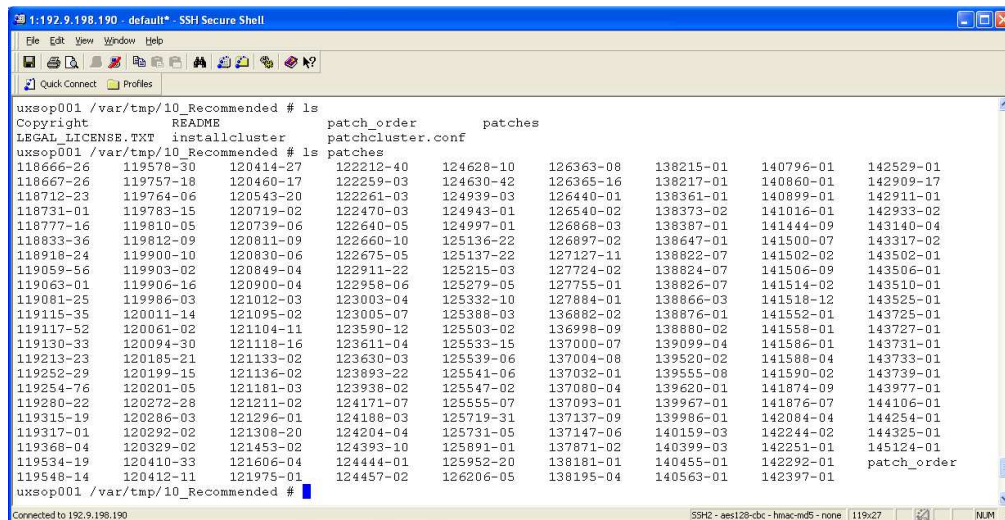


Figura 4.10 Lista de parches

Dentro del conjunto de parches encontraremos cada parche de forma individual, el cual reconocemos por un numero identificador **“patch id”** seguido de un numero de revisión

del parche. Así como un archivo de descripción README en el cual se especifica la forma de instalación y los problemas que corrige.

Para la instalación del conjunto de parches recomendados ejecutamos el archivo ***installcluster***, que instala uno por uno cada parche y si este no es requerido por Solaris 10, no lo instala y continua con el siguiente parche.

Puede ser que no sea necesario instalar cierto parche debido a que se cuenta con una versión actualizada o no contamos con la versión del software el cual presenta el error.

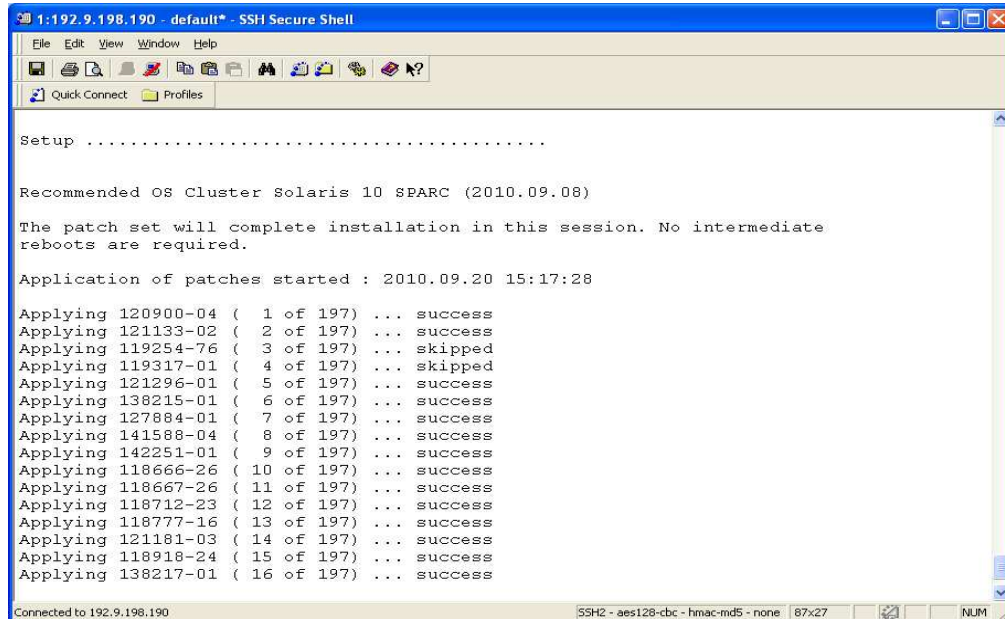


Figura 4.11 Instalación de parches

## Usuarios

Solaris 10 contiene varios usuarios por defecto, se tienen que modificar los privilegios y atributos o en su caso borrar ya que no son necesarias para la ejecución, con el objetivo de fortalecer el aseguramiento del sistema.

Las cuentas de usuarios eliminadas, con el uso del comando ***passmgmt*** con la opción ***-d*** son:

- smtp
- nuucp
- listen

Los usuarios se encuentran eliminados, así como toda referencia a ellos en los archivos ***/etc/passwd*** y ***/etc/shadow***.

El resto de los usuarios por defecto son modificados para agregar más seguridad, ya que no cuentan con un Shell ni con un password asignado. Para evitar que estas cuentas de usuario sean utilizadas se utiliza el comando **passwd** con la opción **-l** para bloquear cada una de ellas. Para impedir que puedan ejecutar comandos se asigna a cada una de las cuentas el Shell **/bin/false**.

La Tabla 4.2 muestra la modificación de usuarios de Sistema Operativo

Usuario	Comentario	Estatus inicial	Estatus modificado	Shell inicial	Shell actual
Daemon	Sin comentario	NL (Non-login)	Bloqueada	Sin Shell	/bin/false
Bin	Sin comentario	NL (Non-login)	Bloqueada	Sin Shell	/bin/false
Sys	Sin comentario	NL (Non-login)	Bloqueada	Sin Shell	/bin/false
Adm	Admin	NL (Non-login)	NL (Non-login)	Sin Shell	/bin/false
Lp	Line Printer Admin	NL (Non-login)	Bloqueada	Sin Shell	/bin/false
Uucp	Uucp Admin	NL (Non-login)	Bloqueada	Sin Shell	/bin/false
Smmsp	Sendmail message Submission Program	NL (Non-login)	Bloqueada	Sin Shell	/bin/false
Gdm	GDM reserved UID	Bloqueada	Bloqueada	Sin Shell	/bin/false
Webservd	WebServer reserved UID	Bloqueada	Bloqueada	Sin Shell	/bin/false
Postgres	PostgreSQL Reserved UID	NL (Non-login)	Bloqueada	/usr/bin/pfksh	/bin/false
Svcstag	Service Tag UID	Bloqueada	Bloqueada	Sin Shell	/bin/false
Nobody	NFS Anonymous Access User	Bloqueada	Bloqueada	Sin Shell	/bin/false
Noaccess	No Access User	Bloqueada	Bloqueada	Sin Shell	/bin/false
Nobody4	SunOS 4.x NFS Anonymous Access User	Bloqueada	Bloqueada	Sin shell	/bin/false

Las cuentas de usuarios con el estatus NL (Non-Login) son cuentas que deben existir en el Sistema Operativo, por ejemplo para reservar un identificador de usuario (UID), sin tener acceso interactivo para la ejecución de comandos. Sin embargo, estas cuentas pueden ejecutar tareas programadas por medio de **cron** y **at** al ser bloqueadas son restringidas por completo.

**Cron** y **At** ejecutan procesos en segundo plano en el Sistema Operativo de forma programada, por lo que cualquier usuario con permisos puede calendarizar la ejecución de un proceso de forma automática. Son de gran ayuda para los administradores del sistema, puesto que ayudan en las tareas de administración.

Pero un usuario no administrativo podría atacar el sistema al dejar corriendo de forma infinita un proceso en segundo plano. Al momento de la instalación de Solaris 10, se crean los archivos *cron.deny* y *at.deny* en la ruta */etc/cron.d/*:

```

uxsop001 /etc/cron.d # ls -lrt
total 6
-rw-r--r--  1 root    sys      17 Jan 21  2005 queuedefs
-rw-r--r--  1 root    sys      40 Sep  3  2009 cron.deny
-rw-r--r--  1 root    sys      40 Sep  3  2009 at.deny
prw-----  1 root    root     0 Oct  9  20:06 FIFO
    
```

Por lo que se agregan a estos archivos los usuarios no administrativos, dejando con permisos de ejecución a “root” y “adm”. Ya que estos ejecutan tareas programadas de administración, importantes para la recolección de información del funcionamiento del equipo.

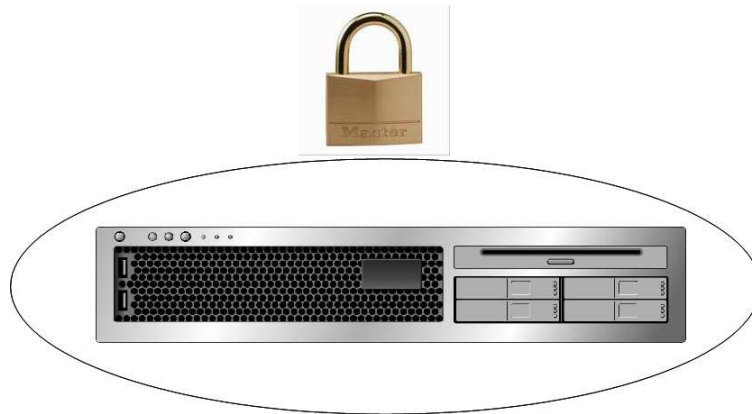


Figura 4.12 Servidor Asegurado

### 4.2.3. Configuración de Espejo de Discos (RAID 1)

Para la configuración de espejo de discos del sistema operativo, se requiere contar con dos discos idénticos en cuanto a tamaño, partición y distribución de espacio en los sistemas de archivos.

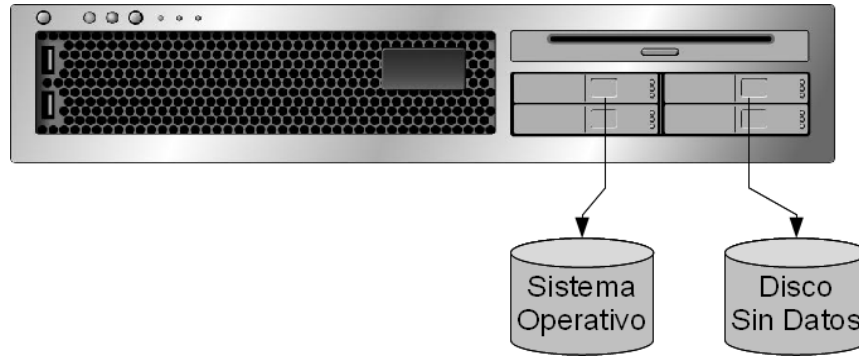


Figura 4.13 Configuración inicial de discos

En primera instancia es necesario crear una base de datos de replicas, las replicas es donde se va almacenar la información y configuración de los dispositivos del espejo. La base de datos se creó utilizando la partición 7 de cada uno de los discos.

```

uxsop001 / # metadb -a -f -c 3 c1t0d0s7 c1t1d0s7
uxsop001 / # metadb -i
      flags          first blk      block count
a m p l u o        16              8192        /dev/dsk/c1t0d0s7
a   p l u o        8208              8192        /dev/dsk/c1t0d0s7
a   p l u o       16400              8192        /dev/dsk/c1t0d0s7
a   p l u o         16              8192        /dev/dsk/c1t1d0s7
a   p l u o        8208              8192        /dev/dsk/c1t1d0s7
a   p l u o       16400              8192        /dev/dsk/c1t1d0s7
r - replica does not have device relocation information
o - replica active prior to last mddb configuration change
u - replica is up to date
l - locator for this replica was read successfully
c - replica's location was in /etc/lvm/mddb.cf
p - replica's location was patched in kernel
m - replica is master, this is replica selected as input
W - replica has device write errors
a - replica is active, commits are occurring to this replica
M - replica had problem with master blocks
D - replica had problem with data blocks
F - replica had format problems
S - replica is too small to hold current data base
R - replica had device read errors
    
```

Se cuenta con dos discos de 146 Gb, c1t0d0 es el primer disco y c1t1d0 es el disco espejo, los cuales se encuentran particionado y se muestran en la tabla 4.3 las Particiones de los discos de arranque.



Partición	Uso	Tamaño
0	/	60.01GB
1	Swap	16.00GB
4	/var	8.01GB
7	bd de replicas	258.38MB

La configuración del espejo se hace convirtiendo cada partición de cada disco en un metadispositivo (un subespejo) mediante el comando **metainit** con la opción **-f**, especificando que es un subespejo de una copia (uno a uno) y la partición del disco. La nomenclatura para nombrar los metadispositivos es la letra “d” seguida del numero de disco (1 ó 2) y la partición a configurar (1, 2 ó 4).

```

uxsop001 / # metainit -f d10 1 1 c1t0d0s0
d10: Concat/Stripe is setup
uxsop001 / # metainit -f d20 1 1 c1t1d0s0
d20: Concat/Stripe is setup
uxsop001 / # metainit -f d11 1 1 c1t0d0s1
d11: Concat/Stripe is setup
uxsop001 / # metainit -f d21 1 1 c1t1d0s1
d21: Concat/Stripe is setup
uxsop001 / # metainit -f d14 1 1 c1t0d0s4
d14: Concat/Stripe is setup
uxsop001 / # metainit -f d24 1 1 c1t1d0s4
d24: Concat/Stripe is setup
    
```

Para las particiones s0, s1 y s4 se hace el espejo, lo que se logra creando los volúmenes d0, d1 y d4. Con el comando **metainit** creamos los subespejos, pero en esta ocasión con la opción **-m** indicando que es un espejo del subespejo del primer disco.

```

uxsop001 / # metainit d0 -m d10
d0: Mirror is setup
uxsop001 / # metainit d1 -m d11
d1: Mirror is setup
uxsop001 / # metainit d4 -m d14
d4: Mirror is setup
    
```

Una vez configurado el espejo se edita el archivo **/etc/vfstab**, que contiene la información de los sistemas de archivos que debe montar el sistema operativo al momento de arrancar. Se cambia el punto de montura de las particiones por el metadispositivo espejo creado.

```

uxsop001 / # cat /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type    pass     at boot   options
#
fd -         /dev/fd fd -       no -
/proc -      /proc proc -       no -
/dev/md/dsk/d1 - - swap -       no -
/dev/md/dsk/d0 /dev/md/rdsk/d0 / ufs 1 no -
/dev/md/dsk/d4 /dev/md/rdsk/d4 /var/crash ufs 2 yes -
/devices -     /devices devfs -       no -
sharefs -    /etc/dfs/sharetab sharefs -     no -
ctfs -      /system/contract ctfs -       no -
objfs -     /system/object objfs -       no -
swap -      /tmp tmpfs -      yes -
    
```

Es necesario reiniciar el sistema operativo para que levante los nuevos metadispositivos espejo. Una vez validado, se procede a juntar los subespejos del disco espejo.

```

uxsop001 / # metattach d0 d20
uxsop001 / # metattach d1 d21
uxsop001 / # metattach d4 d24
    
```

Finalmente con el comando **metastat** se monitorea el proceso de sincronización del primer disco al disco espejo, quedando configurado el espejo para los discos de arranque (Figura 4.14).

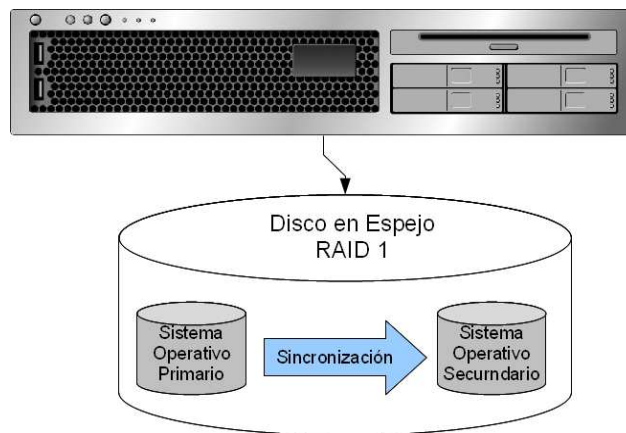


Figura 4.14 Creación del Espejo Final

### 4.3. Configuración de Solaris Containers

Este subtema abarca la configuración de los Solaris Containers, tema principal de este trabajo. En el desarrollo del subtema se muestra un resumen de la configuración de los Solaris Containers, el detalle de toda la configuración se encuentra en el Anexo 3.

La configuración de los Solaris Containers implica ciertos pasos, los cuales se desarrollan más adelante en este subtema y son:

- Configuración del Pool de Recursos
- Definición de Zonas

#### 4.3.1. Configuración Pool de recursos

Las zonas ya sean globales o locales tienen asociado un *Resource Pool*, que es una entidad lógica que contiene recursos del sistema, como lo son CPU y memoria. Este Resource Pool puede ser independiente o estar de forma compartida con una o varias zonas.

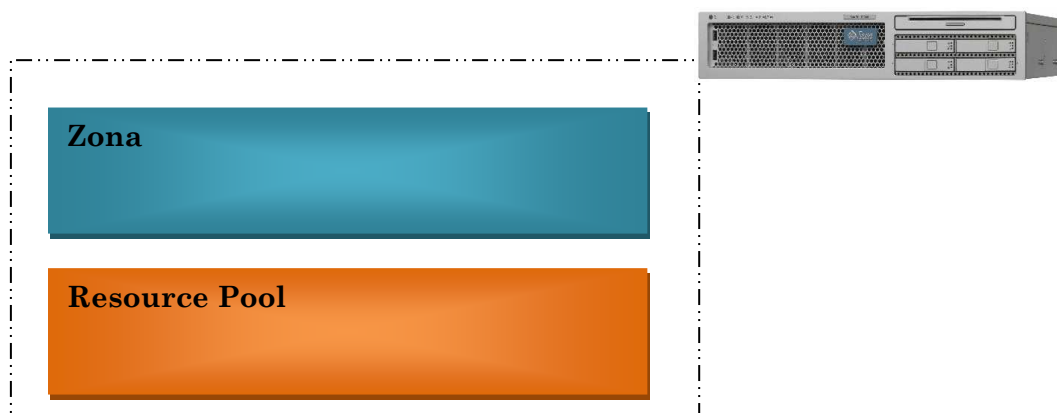


Figura 4.15 Configuración de Solaris Container

A continuación describimos brevemente la configuración y creación de los pool's de recursos para el Servidor Aplicativo de Colaboración.

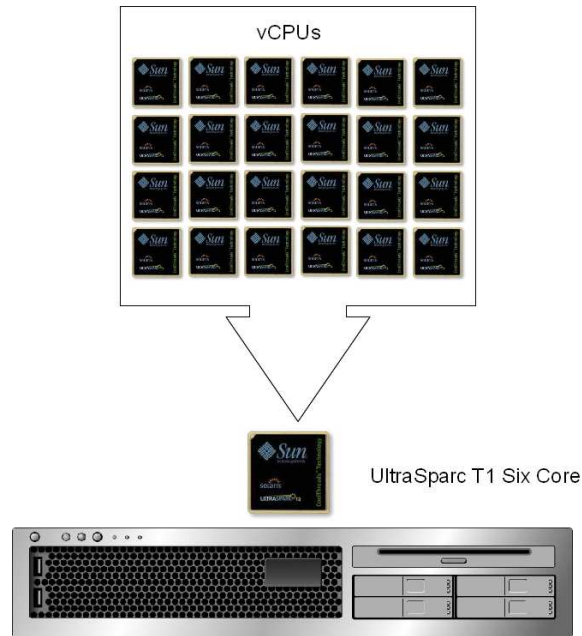


Figura 4.16 Pool de Recursos Inicial

El demonio `poold` es el encargado de controlar y proveer facilidades de particionamiento de recursos a la hora de crear los Pool's de Recursos. Para poder llevar a cabo la configuración de los Pool's de Recursos para las zonas que vamos a crear, es necesario verificar que antes de iniciar la configuración, el demonio `poold` esté activado, si no está activado se debe proceder a habilitarlo.

Verificamos que el demonio `poold` se encuentre activado mediante el comando `pgrep` y el parámetro `-lf poold`, posteriormente verificamos si los servicios se encuentran habilitados mediante el comando `svcs`.

```

uxsop001 / # pgrep -lf poold
uxsop001 / #
uxsop001 / # svcs -a |grep pools
disabled      Oct_23   svc:/system/pools/dynamic:default
online        Oct_24   svc:/system/pools:default
    
```

Se puede observar que el servicio de "Resource Pool" se encuentra habilitado, mientras que el servicio de "Resource Pool Dinámico" no lo está, ya que este servicio es dependiente del primero, para habilitarlo utilizamos el comando `svcadm` con los parámetros `enable -r`, especificando que habilitaremos los dos servicios:

```

uxsop001 / #
uxsop001 / # svcadm enable -r pools/dynamic
uxsop001 / #
    
```

Una vez habilitados los servicios verificamos su estado mediante el comando **svcs**. En la columna de estado observamos que ambos servicios se encuentran habilitados.

```

uxsop001 / # svcs -a |grep pools
online      13:30:18 svc:/system/pools/dynamic:default
online      Oct_24   svc:/system/pools:default
    
```

Nuevamente obtenemos el estado del demonio *poold* mediante **pgrep** y observamos que el demonio se encuentra activo.

```

uxsop001 / # pgrep -lf poold
28467 /usr/lib/pool/poold
    
```

Para la configuración de los Pool's, se usa el comando **poolcfg**, el cual provee todas las operaciones necesarias para esta tarea. Inicialmente utilizando el comando **poolcfg** con el argumento **-c**, creamos el primer conjunto de procesadores (*pset*) que en nuestro caso contendrá 2 vCPU como recurso de procesamiento para el contenedor de Servidor de Sitio de Colaboración.

Después de crear el conjunto de procesadores, se deben de especificar como parámetros el número de procesadores virtuales (vCPU's) que va a tener el *pset* recién creado para el Servidor del Sitio de Colaboración y es por ello que especificamos que máximo tiene 2 y mínimo tiene 2.

También se le define el nombre, para nuestro caso el nombre es colaboración-pset. Posteriormente se crea el pool de recursos para este *pset* mediante el comando **poolcfg** y el argumento **-c**.

```

uxsop001 / #poolcfg -c 'create pset pset_collabapp'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pset pset_collabapp (unit pset.min = 2;
unit pset.max = 2)'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pool pool_collabapp'
    
```

Una vez creados el *pset* y el *pool*, se requiere crear un vínculo entre ambos. Para este proceso se utiliza **associate pool** y posteriormente se activa esta configuración mediante el comando **pooladm -c**.

```
uxsop001 / #poolcfg -c 'associate pool pool_collabapp (pset pset_collabapp)'
uxsop001/ #
```

En este momento ya tenemos creado el pool de recursos para el Servidor del Sitio de Colaboración, solamente verificamos la existencia del nuevo pool utilizando el comando **poolcfg -dc info**.

```
uxsop001 / # poolcfg -dc info

system default
  string system.comment
  int system.version 1
  boolean system.bind-default true
  string system.poold.objectives wt-load

pool pool_default
  int pool.sys_id 0
  boolean pool.active true
  boolean pool.default true
  int pool.importance 1
  string pool.comment
  pset pset_default

pool pool_collabapp
  int pool.sys_id 8
  boolean pool.active true
  boolean pool.default false
  int pool.importance 1
  string pool.comment
  pset pset_collabapp

pset pset_collabapp
  int pset.sys_id 1
  boolean pset.default false
  uint pset.min 2
  uint pset.max 2
  string pset.units population
  uint pset.load 0
  uint pset.size 2
  string pset.comment

cpu
  int cpu.sys_id 17
  string cpu.comment
  string cpu.status on-line

cpu
  int cpu.sys_id 16
  string cpu.comment
  string cpu.status on-line
```

```

cpu
    int    cpu.sys_id 17
    string cpu.comment
    string cpu.status on-line

cpu
    int    cpu.sys_id 16
    string cpu.comment
    string cpu.status on-line

```

Enseguida se configuran y crean los pool's de recursos para los demás contenedores.

### Creación del POOL de Recursos Collab DB

```

uxsop001 / #poolcfg -c 'create pset pset_collabdb'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pset pset_collabdb (unit pset.min = 2; unit
pset.max = 2) '
uxsop001 / #
uxsop001 / #poolcfg -c 'create pool pool_collabdb'
uxsop001 / #
uxsop001 / #poolcfg -c 'associate pool pool_collabdb (pset pset_collabdb) '

```

### Creación del POOL de Recursos Joomla App

```

uxsop001 / #poolcfg -c 'create pset pset_joomlaapp'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pset pset_joomlaapp (unit pset.min = 4; unit
pset.max = 4) '
uxsop001 / #
uxsop001 / #poolcfg -c 'create pool pool_joomlaapp'
uxsop001 / #
uxsop001 / # poolcfg -c 'associate pool pool_joomlaapp (pset pset_joomlaapp) '

```

### Creación del POOL de Recursos Joomla DB

```

uxsop001 / #poolcfg -c 'create pset pset_joomladb'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pset pset_joomladb (unit pset.min = 2; unit
pset.max = 2) '
uxsop001 / #
uxsop001 / #poolcfg -c 'create pool pool_joomladb'
uxsop001 / #
uxsop001 / #poolcfg -c 'associate pool pool_joomladb (pset pset_joomladb) '

```

### Creación del POOL de Recursos Ecommerce DB

```

uxsop001 / #poolcfg -c 'create pset pset_ecommercedb'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pset pset_ecommercedb (unit pset.min = 2; unit
pset.max = 2) '
uxsop001 / #
uxsop001 / #poolcfg -c 'create pool pool_ecommercedb'
uxsop001 / #
uxsop001 / #poolcfg -c 'associate pool pool_ecommercedb (pset
pset_ecommercedb) '
    
```

### Creación del POOL de Recursos Ecommerce App

```

uxsop001 / #poolcfg -c 'create pset pset_ecommerceapp'
uxsop001 / #
uxsop001 / #poolcfg -c 'create pset pset_ecommerceapp (unit pset.min = 8;
unit pset.max = 8) '
uxsop001 / #
uxsop001 / #poolcfg -c 'create pool pool_ecommerceapp'
uxsop001 / #
uxsop001 / #poolcfg -c 'associate pool pool_ecommerceapp (pset
pset_ecommerceapp) '
    
```

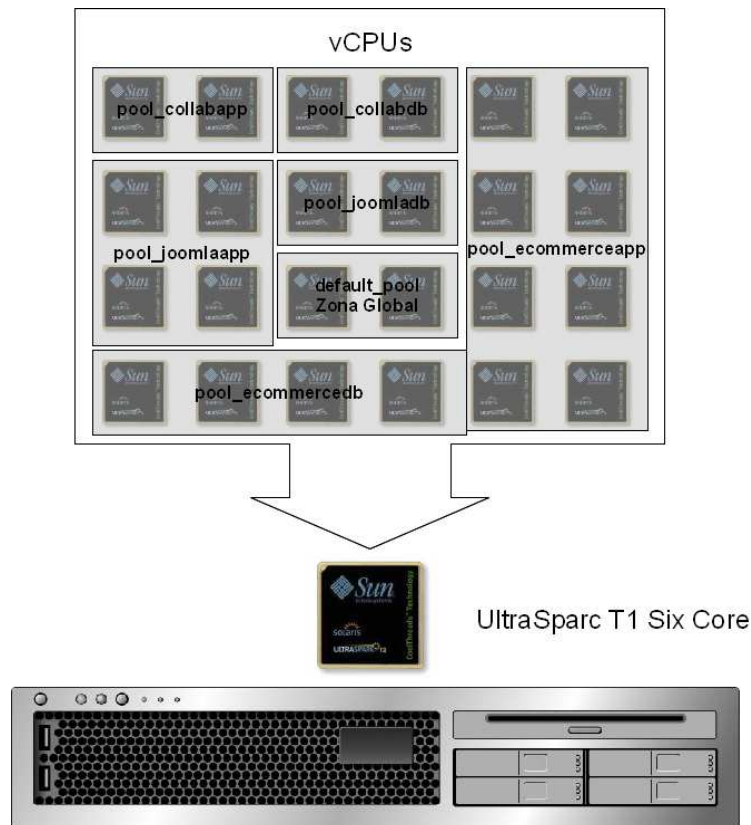


Figura 4.17 Pool de Recursos Final



#### 4.4. Definición Zonas

Como se había mencionado en el capítulo 2, el sistema operativo base Solaris 10 contiene la Zona Global que se encarga de administrar el uso y acceso de los recursos, de la misma forma que permite crear, instalar y administrar las Zonas Locales o Zonas no Globales.

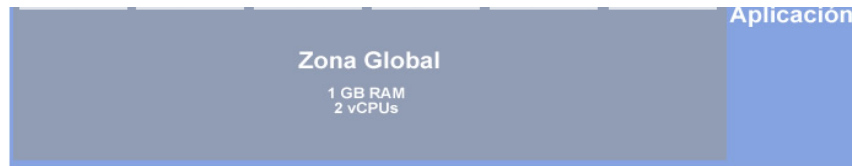


Figura 4. 18 Zona Global

En esta sección explicaremos brevemente la creación, instalación y administración de dichas Zonas Locales, las cuales son del tipo **whole root zone**, esto con la finalidad de proveer independencia a cada zona, ya que en las de tipo **sparse root zone** se instalan solamente ciertos directorios y hereda por default otros de solo lectura de la zona global como son **/lib**, **/platform**, **/sbin** y **/usr** y como consecuencia son dependientes de la zona global.

#### Descripción de la creación de la Zona Joomla Aplicación

Las zonas locales tienen un filesystem del tipo ZFS, es por ello que en primera instancia creamos este sistema de archivos mediante el comando `create`, el cual requiere como parámetro el directorio del pool y el nombre de la zona, en nuestro caso `zones` y con el nombre de `joomla-app` respectivamente. Restringimos el tamaño que ocupará la zona en el disco mediante **set quota**, se le establece como tamaño 10 GB, espacio suficiente para la instalación y el buen desempeño de la aplicación.

```

uxsop001 / #zfs create zones/joomla-app
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/joomla-app
    
```

Una vez creado el sistema de archivos utilizamos el comando **zonecfg** con el parámetro **-z** y definimos el nombre de la zona a crear, para nuestro caso **joomla-app**.

Este comando nos indica que no existe la zona configurada con la leyenda: “*No such zone configured*”; pero abre el Shell **zonecfg:joomla-app**.

El Shell nos permite la creación de las zonas mediante el uso del comando **create** con el parámetro **-b**, para indicar que se trata de una nueva zona de tipo **whole root zone**.

```
uxsop001 / #zonecfg -z joomla-app
joomla-app: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:joomla-app> create -b
```

Una vez creada la zona la estableceremos en la ruta del directorio raíz con el comando **set zonepath**: El directorio de la zona local debe tener permisos del modo 755 (todos los privilegios para el usuario root, lectura y ejecución para otros y el grupo), ya que de esta forma se asegura que los usuarios sin privilegios de la zona global no pasen a un sistema de archivos de una zona local. Adicionalmente se debe especificar si la zona tendrá el parámetro de **autoboot activado**, en nuestro caso le configuramos que **true**, esto para que de forma automática reinicie.

```
uxsop001 / #zonecfg -z joomla-app
joomla-app: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:joomla-app> create -b
zonecfg:joomla-app>
zonecfg:joomla-app> set autoboot=true
zonecfg:joomla-app> set zonepath=/zones/joomla-app
```

Los parámetros de red los especificamos con el comando **add net**, una vez dentro del Shell definimos la dirección IP que se va a utilizar mediante **set address**, seguido de la IP **192.9.198.213**. Posteriormente le asignamos dicha IP a la interfaz de red **e1000g0** mediante **set physical** y finalizamos la asignación con **end**.

```

uxsop001 / #zonecfg -z joomla-app
joomla-app: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:joomla-app> create -b
zonecfg:joomla-app>
zonecfg:joomla-app> set autoboot=true
zonecfg:joomla-app> set zonepath=/zones/joomla-app
zonecfg:joomla-app>
zonecfg:joomla-app> add net
zonecfg:joomla-app:net>
zonecfg:joomla-app:net> set address=192.9.198.213
zonecfg:joomla-app:net> set physical=e1000g0
zonecfg:joomla-app:net> end

```

El comando **verify** se utiliza para verificar si sintácticamente es correcta la configuración, podemos observar que al no obtener ningún error la zona.

Finalmente con el comando **commit**, se mueve la configuración de la memoria a disco. Para finalizar el proceso tecleamos **exit**.

```

uxsop001 / #zonecfg -z joomla-app
joomla-app: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:joomla-app> create -b
zonecfg:joomla-app>
zonecfg:joomla-app> set autoboot=true
zonecfg:joomla-app> set zonepath=/zones/joomla-app
zonecfg:joomla-app>
zonecfg:joomla-app> add net
zonecfg:joomla-app:net>
zonecfg:joomla-app:net> set address=192.9.198.213
zonecfg:joomla-app:net> set physical=e1000g0
zonecfg:joomla-app:net> end
zonecfg:joomla-app>
zonecfg:joomla-app> verify
zonecfg:joomla-app> commit
zonecfg:joomla-app> exit

```

La instalación se lleva a cabo a través del comando **zoneadm** con la opción **z** seguido del nombre de la zona y después **install**, el cual verifica la disponibilidad de los recursos y además crea los puntos de montajes e instala la zona. Durante este proceso se muestran mensajes del avance los archivos y directorios instalados, al igual que informa errores en caso de que estos ocurran.

```

uxsop001 / # zoneadm -z colaboracion-app install
cannot create ZFS dataset zones/colaboracion-app: dataset already exists
Preparing to install zone <colaboracion-app>.
Creating list of files to copy from the global zone.
Copying <148832> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1205> packages on the zone.
Initialized <1205> packages on zone.
Zone <colaboracion-app> is initialized.
Installation of <4> packages was skipped.
Installation of these packages generated warnings: <SUNWtcatr>
The file </zones/colaboracion-app/root/var/sadm/system/logs/install_log> contains a log of the zone
installation.
uxsop001 / #

```

La primera zona fue creada e instalada con éxito, no se obtuvo ningún error por lo procedemos a la creación y configuramos el resto de las zonas.

### Configuración y Creación de la zona Joomla-DB

```

uxsop001 / #zfs create zones/joomla-db
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/joomla-db
uxsop001 / #
uxsop001 / #zonecfg -z joomla-db
joomla-db: No such zone configured
Use 'create' to begin configuring a new zone.

```

```

uxsop001 / #zfs create zones/joomla-db
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/joomla-db
uxsop001 / #
uxsop001 / #zonecfg -z joomla-db
joomla-db: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:joomla-db> create -b
zonecfg:joomla-db>
zonecfg:joomla-db> set autoboot=true
zonecfg:joomla-db> set zonepath=/zones/joomla-db
zonecfg:joomla-db>
zonecfg:joomla-db> add net
zonecfg:joomla-db:net>
zonecfg:joomla-db:net> set address=192.9.198.212
zonecfg:joomla-db:net> set physical=e1000g0
zonecfg:joomla-db:net> end
zonecfg:joomla-db>
zonecfg:joomla-db> verify
zonecfg:joomla-db> commit
zonecfg:joomla-db> exit

```

### Configuración y Creación de la zona Colaboración-APP

```

uxsop001 / #zfs create zones/colaboracion-app
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/colaboracion-app
uxsop001 / #
uxsop001 / #zonecfg -z colaboracion-app
colaboracion-app: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:colaboracion-app> create -b
zonecfg:colaboracion-app>
zonecfg:colaboracion-app> set autoboot=true
zonecfg:colaboracion-app> set zonepath=/zones/colaboracion-app
zonecfg:colaboracion-app>
zonecfg:colaboracion-app> add net
zonecfg:colaboracion-app:net>
zonecfg:colaboracion-app:net> set address=192.9.198.185
zonecfg:colaboracion-app:net> set physical=e1000g0
zonecfg:colaboracion-app:net> end
zonecfg:colaboracion-app>
zonecfg:colaboracion-app> verify
zonecfg:colaboracion-app> commit
zonecfg:colaboracion-app> exit

```

### Configuración y Creación de la zona Colaboración-DB

```

uxsop001 / #zfs create zones/colaboracion-db
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/colaboracion-db

```

```

uxsop001 / #zonecfg -z colaboracion-db
colaboracion-db: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:colaboracion-db> create -b
zonecfg:colaboracion-db>
zonecfg:colaboracion-db> set autoboot=true
zonecfg:colaboracion-db> set zonepath=/zones/colaboracion-db
zonecfg:colaboracion-db>
zonecfg:colaboracion-db> add net
zonecfg:colaboracion-db:net>
zonecfg:colaboracion-db:net> set address=192.9.198.91
zonecfg:colaboracion-db:net> set physical=e1000g0
zonecfg:colaboracion-db:net> end
zonecfg:colaboracion-db>
zonecfg:colaboracion-db> verify
zonecfg:colaboracion-db> commit
zonecfg:colaboracion-db> exit

```

### Configuración y Creación de la zona Ecommerce-DB

```

uxsop001 / #zfs create zones/ecommerce-db
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/ecommerce-db
uxsop001 / #
uxsop001 / #zonecfg -z ecommerce-db
ecommerce-db: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:ecommerce-db> create -b
zonecfg:ecommerce-db>
zonecfg:ecommerce-db> set autoboot=true
zonecfg:ecommerce-db> set zonepath=/zones/ecommerce-db
zonecfg:ecommerce-db>
zonecfg:ecommerce-db> add net
zonecfg:ecommerce-db:net>
zonecfg:ecommerce-db:net> set address=192.9.198.71
zonecfg:ecommerce-db:net> set physical=e1000g0
zonecfg:ecommerce-db:net> end
zonecfg:ecommerce-db>
zonecfg:ecommerce-db> verify
zonecfg:ecommerce-db> commit
zonecfg:ecommerce-db> exit

```

### Configuración y Creación de la zona Ecommerce-App

```

uxsop001 / #zfs create zones/ecommerce-app
uxsop001 / #
uxsop001 / #zfs set quota=10g zones/ecommerce-app
uxsop001 / #
uxsop001 / #zonecfg -z ecommerce-app
ecommerce-app: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:ecommerce-app> create -b
zonecfg:ecommerce-app>
zonecfg:ecommerce-app> set autoboot=true
zonecfg:ecommerce-app> set zonepath=/zones/ecommerce-app
zonecfg:ecommerce-app>
zonecfg:ecommerce-app> add net
zonecfg:ecommerce-app:net>
zonecfg:ecommerce-app:net> set address=192.9.198.235
zonecfg:ecommerce-app:net> set physical=e1000g0
zonecfg:ecommerce-app:net> end
zonecfg:ecommerce-app>
zonecfg:ecommerce-app> verify
zonecfg:ecommerce-app> commit
zonecfg:ecommerce-app> exit

```

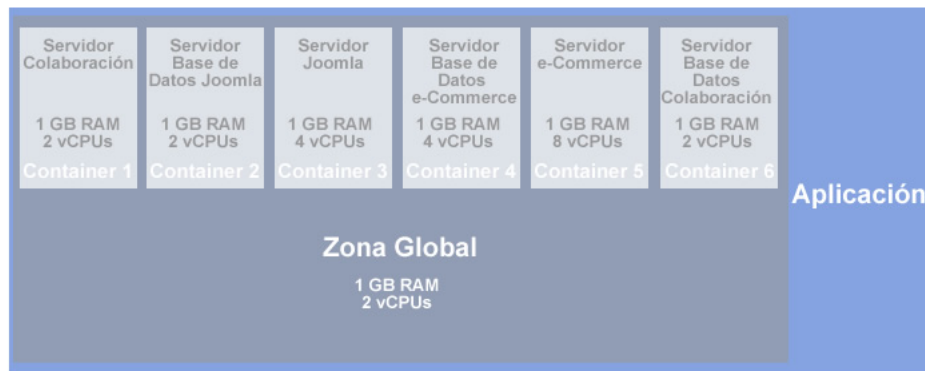


Figura 4.18 Configuración Final Zona Global y zonas locales

Ya que se tienen creadas las zonas (Figura 4.18), procedemos a la instalación de cada una de ellas. Mediante el comando **zoneadm list** y el parámetro **-cv** obtenemos la información de las zonas, la primera columna indica el “*ID Name*” que es el nombre de la zona, la siguiente columna indica el estado, aquí podemos observar que todas las zonas se encuentran instaladas y que tienen una dirección IP del tipo *Shared*.

```

uxsop001 / # zoneadm list -cv
ID NAME          STATUS    PATH                                BRAND  IP
0 global          running  /                                  native shared
9 colaboracion-app installed /zones/colaboracion-app          native shared
10 colaboracion-db installed /zones/colaboracion-db          native shared
11 ecommerce-db   installed /zones/ecommerce-db            native shared
12 ecommerce-app  installed /zones/ecommerce-app           native shared
13 joomla-db      installed /zones/joomla-db               native shared
14 joomla-app     installed /zones/joomla-app              native shared
    
```

Después de realizar este procedimiento, continuamos a iniciar sesión en cada zona con la finalidad de realizar la instalación de Solaris, la cual se realiza de forma muy similar a la instalación del Sistema Operativo base y con los mismos parámetros de configuración. Se inicia sesión con el comando **zlogin -C** y el nombre de la zona. Posteriormente seleccionamos el idioma, en nuestro caso inglés que es la opción **0** (Figura 4.19), adicionalmente especificamos en **locale** la opción **0** que es la adecuada para el teclado, ya que la otras opciones son para sitios geográficos en Canadá y USA:

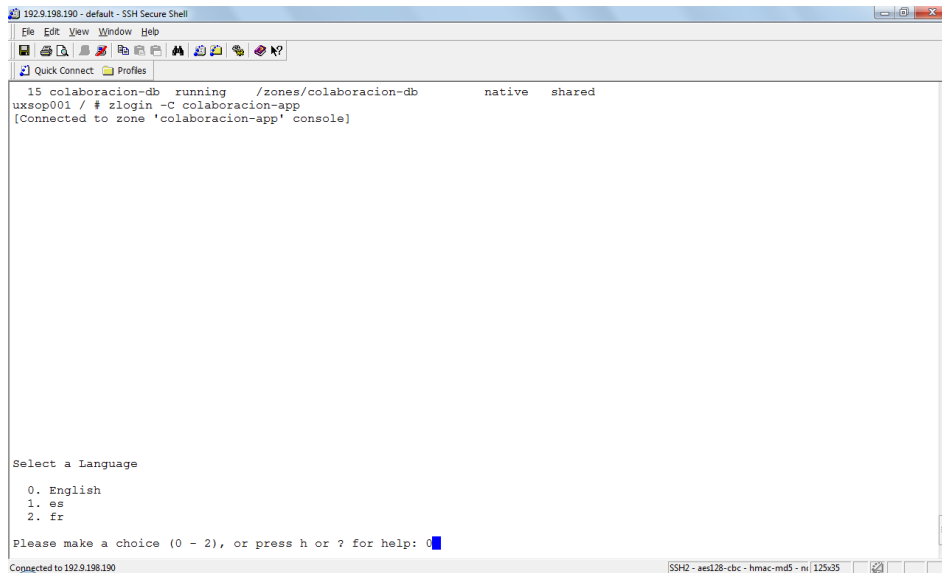


Figura 4.19 Conexión a consola de la zona colaboración-app

En la siguiente pantalla seleccionamos el tipo de terminal que estamos usando, en nuestra instalación se utilizó la opción **3**, correspondiente a DEC VT100.

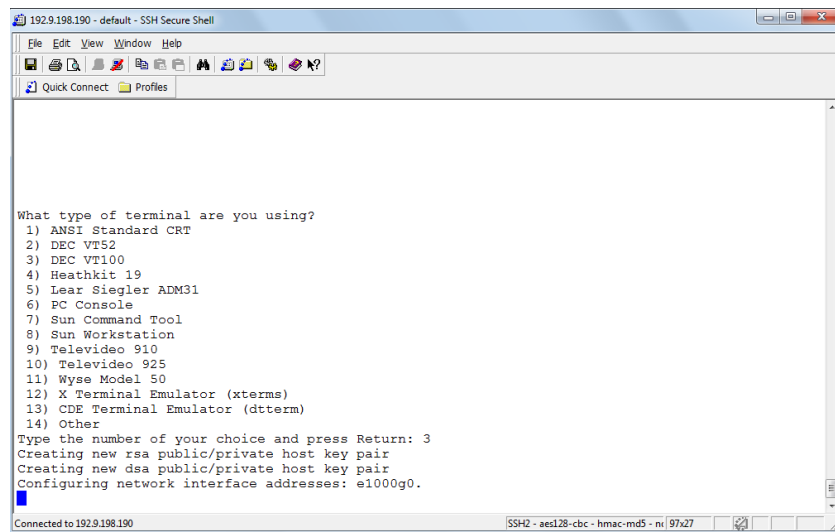


Figura 4.20 Configuración inicial de la zona colaboración-app

Una vez hecho esto especificamos los parámetros de la configuración como son el nombre del host, políticas de seguridad de kerberos, servicios habilitados, etc., las configuraciones para cada contenedor se detallan en el Anexo 3 de este documento.



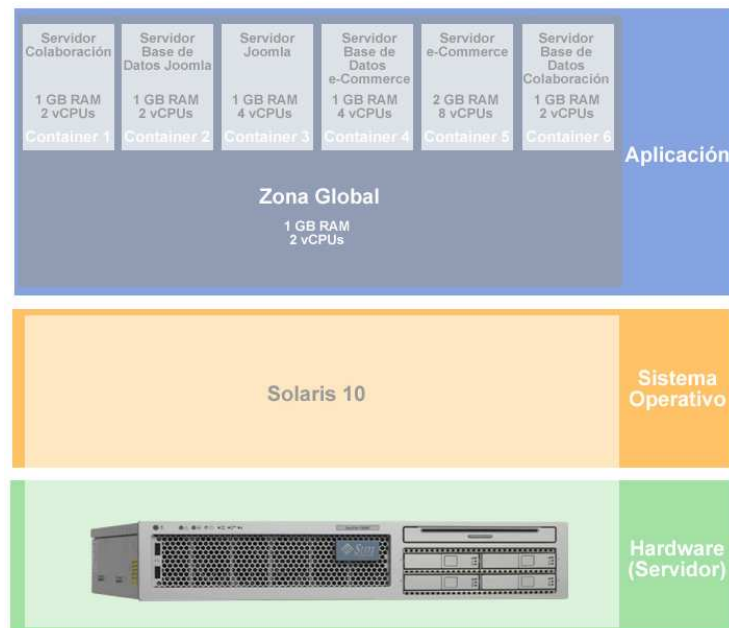
## Asignación de Memoria RAM

Para la asignación de memoria RAM para las zonas creadas, decidimos usar la técnica llamada “*Memory Capping*”, la cual permite establecer un límite máximo a los recursos de memoria física. Tanto en Zonas como en Proyectos de Solaris 10, la definición de límites de memoria física es posible.

Para este trabajo de Tesis, decidimos usar la técnica de “*Memory Capping*” ya que de esta manera se establece un control administrado del uso de memoria por cada Zona, es decir, cada Zona puede trabajar con el máximo de memoria física establecida sin agotar los recursos de memoria de otras zonas. Cuando se decide usar la técnica “*Memory Capping*”, se debe de tener muy clara la cantidad de RAM que necesita cada aplicativo que albergará una Zona, ya que si se establece un límite y éste no satisface las necesidades del aplicativo, se tendrán problemas de rendimiento y el aplicativo no funcionará.

El demonio *rcapd* es el encargado de proporcionar los mecanismos necesarios para la asignación y administración de los límites de memoria ó “*Memory Capping*”.

La asignación de memoria por Zona, se definió en el capítulo anterior y se muestra en la figura 4.21:



4.21 Detalle de Memoria Física por Contenedor

A continuación se describe brevemente la configuración de “*Memory Capping*” para la Zona *colaboracion-db*:

Primero, se tiene que verificar que el servicio *rcap* se encuentre activo, de otra manera no se podrá realizar la configuración. Con el comando **svcadm** y el argumento **enable** habilitamos el servicio *rcap*:

```
uxsop001# svcadm enable rcap
```

Para ejecutar estos comandos, se debe de contar con privilegios de super usuario del sistema. Para que el demonio *rcapd* inicie inmediatamente y también inicialice cada vez que el sistema inicia, se usa el comando **rcapadm** con el argumento **-E**:

```
uxsop001# rcapadm -E
```

Una vez que habilitamos el servicio *rcap*, podemos comenzar con la configuración de los límites de memoria para las zonas.

El detalle de configuración de memoria para todas las zonas, se muestra en el Anexo 4.

A continuación se muestra la configuración para la zona *colaboracion-db*.

```
uxsop001# zonecfg -z colaboración-db
zonecfg:colaboracion-db> add capped-memory
zonecfg:colaboración-db:capped-memory> set physical=1g
zonecfg:colaboracion-db:capped-memory> end
zonecfg:colaboracion-db> verify
zonecfg:colaboración-db> commit
zonecfg:colaboración-db> exit
```

Para finalizar la configuración, debe de reiniciarse la zona:

```
uxsop001# zlogin colaboración-db init 6
```

El mismo procedimiento debe de realizarse con el resto de las Zonas.

#### 4.5. Instalación de Webstack

Posterior a la instalación y configuración del Sistema Operativo Base, la generación y configuración de los Contenedores, es necesario instalar el Software que servirá para dar servicio, tal como se ha planteado en la sección de Diseño del presente documento.

Para este trabajo de Tesis se ha usado un producto que cumple con los requerimientos de los aplicativos descritos, llamado Webstack, producto de la empresa Sun Microsystems, que integra un servidor web Apache, un intérprete de PHP compatible y pre-configurado para funcionar con Apache; y por último MySQL. Esta combinación es conocida como un Stack AMP (Apache MySQL PHP), así mismo es muy común su uso para aplicaciones web. La ventaja de usar Sun Webstack, en lugar de otros productos que integran los mismos elementos de software, es que fue desarrollado por la misma empresa que el sistema operativo, con ello se reducen riesgos por compatibilidad así mismo tiene una mejor integración.

#### **4.5.1. Servidor de Aplicaciones**

Para el caso de un servidor basado en el Stack AMP, un Servidor de aplicaciones está compuesto por el servidor Web Apache y el intérprete de PHP. El primero es el que recibe las peticiones desde los clientes remotos, pasándolas al intérprete de PHP, quién, dependiendo de cómo haya sido concebida la aplicación, puede generar contenido dinámico o interactuar con la base de datos, regresando los datos procesados al servidor Web para ser mostrados al usuario.

La instalación de Apache y de PHP se realiza desde los archivos de instalación de Sun Webstack, que al ser integrado por el mismo Sun Microsystems reduce la operación a un par de comandos, uno instala el software y el otro lo activa.

La instalación de cualquier software en los Solaris Containers requiere que se esté dentro del Shell del contenedor, esto se logra con el comando **zlogin** y el nombre de la Zona desde la zona global (o sistema operativo base), el procedimiento se muestra en la figura 4.22.

```

192.9.198.190 - PuTTY
uxsop001 / # zoneadm list -cv
ID NAME STATUS PATH BRAND IP
0 global running / native shared
15 colaboracion-db running /zones/colaboracion-db native shared
16 colaboracion-app running /zones/colaboracion-app native shared
17 ecommerce-app running /zones/ecommerce-app native shared
18 joomla-app running /zones/joomla-app native shared
19 joomla-db running /zones/joomla-db native shared
20 ecommerce-db running /zones/ecommerce-db native shared
uxsop001 / # zlogin ecommerce-app
[Connected to zone 'ecommerce-app' pts/6]
Last login: Sun Oct 31 13:38:05 on console
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
You have new mail.
# bash
bash-3.00# mkdir /export/software
bash-3.00# mount -F nfs 192.9.198.190:/export/software/webstack /export/software/
bash-3.00# ls /export/software/
README sun-mysql150.pkg
SUNWpsn.pkg sun-mysql150test.pkg
SUNWservicetagr.pkg sun-php52-apc.pkg
SUNWservicetagu.pkg sun-php52-dtrace.pkg
SUNWsthvreg.pkg sun-php52-idn.pkg
SUNWstosreg.pkg sun-php52-memcache.pkg
data sun-php52-suhosin.pkg
install sun-php52-tcpwrap.pkg
legal sun-php52-xdebug.pkg
sun-apache22-dtrace.pkg sun-php52.pkg
sun-apache22-fcgid.pkg sun-python25.pkg
sun-apache22-jk.pkg sun-ruby18.pkg
sun-apache22-perl.pkg sun-squid.pkg
sun-apache22.pkg sun-wsbase.pkg
sun-lighttpd14.pkg webstack-native-1.4-b06-solaris-sparc.tar
sun-memcached.pkg
bash-3.00#

```

Figura 4.22 Conexión a la zona ecommerce-app

En la figura 4.23 se muestra la instalación de Apache y PHP mediante el instalador de Webstack, se asume que los archivos de instalación se encuentran en el Contenedor correspondiente. Desde el directorio donde se encuentran los archivos de instalación se ingresa el comando **.install php** y posteriormente **.install apache**, esta acción copia los archivos ejecutables o binarios, asimismo inicializa los archivos de configuración del Sistema Operativo.

```

192.9.198.190 - PuTTY
bash-3.00# cd /export/software/
bash-3.00# ls
README sun-mysql150.pkg
SUNWpsn.pkg sun-mysql150test.pkg
SUNWservicetagr.pkg sun-php52-apc.pkg
SUNWservicetagu.pkg sun-php52-dtrace.pkg
SUNWsthvreg.pkg sun-php52-idn.pkg
SUNWstosreg.pkg sun-php52-memcache.pkg
data sun-php52-suhosin.pkg
install sun-php52-tcpwrap.pkg
legal sun-php52-xdebug.pkg
sun-apache22-dtrace.pkg sun-php52.pkg
sun-apache22-fcgid.pkg sun-python25.pkg
sun-apache22-jk.pkg sun-ruby18.pkg
sun-apache22-perl.pkg sun-squid.pkg
sun-apache22.pkg sun-wsbase.pkg
sun-lighttpd14.pkg webstack-native-1.4-b06-solaris-sparc.tar
sun-memcached.pkg
bash-3.00# zonename
ecommerce-app
bash-3.00# ./install php
WS104 & browser could not be invoked to display the Product Registration page. You may register the product
by opening the following URL in a browser:
file:///tmp/webstack_register.root.html

Installation was successful.
bash-3.00# ./install apache
WS104 & browser could not be invoked to display the Product Registration page. You may register the product
by opening the following URL in a browser:
file:///tmp/webstack_register.root.html

Installation was successful.
bash-3.00#

```

Figura 4.23 Instalación de Apache y PHP

Posterior a la instalación, es necesario inicializar el servicio, debido a la integración de Webstack con Solaris, esto se limita al comando **svcadm enable sun-apache22**, de otra forma, la secuencia de comandos depende del sistema operativo y de cada una de las compilaciones de Apache y PHP. Para inicializar y validar que Apache está activo, se utiliza el comando **svcs | grep apache**. Adicional a la activación, es necesario verificar la dirección IP correspondiente a la Zona (Figura 4.24).

```

192.9.198.190 - PuTTY
bash-3.00# svcs | grep apache
legacy_run      13:34:11 lrc:/etc/rc3_d/$S0apache
bash-3.00# svcadm enable sun-apache22
bash-3.00# svcs | grep apache
legacy_run      13:34:11 lrc:/etc/rc3_d/$S0apache
offline*        20:46:31 svc:/network/http:sun-apache22
bash-3.00# ifconfig -a
lo0:4: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ffffffff
e1000g0:5: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.9.198.235 netmask ffffffff0 broadcast 192.9.198.255
bash-3.00#
    
```

Figura 4.24 Configuración de red de la zona

Para validar que el servicio del servidor de aplicativos y web está activo, Webstack instala datos de ejemplo, los cuales pueden ser visualizados desde un navegador web, tecleando como dirección la dirección IP (Figura 4.25).

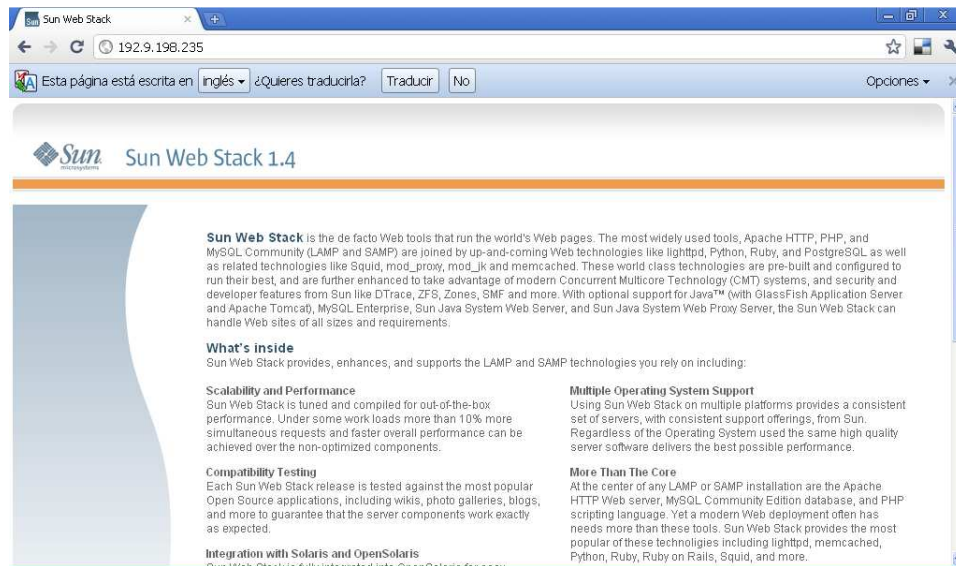


Figura 4.25 Pantalla webstack

#### 4.5.2. Servidor de Base de datos

Un servidor de Base de Datos, es un sistema informático cuya función es la de ser un repositorio de datos, en particular son datos que residen en estructuras llamadas tablas.

Estos servidores son capaces de manipular y administrar dichas estructuras y la información que en ellas residen, además administran el acceso a la información.

Para este trabajo se usó el motor de base de datos de MySQL, debido a que forma parte de Webstack, por ello tiene una mejor integración con el entorno completo. Además MySQL es la base de datos en la que fueron desarrollados los aplicativos que estarán albergados en el servidor del trabajo de Tesis(Joomla, E-Commerce y Collabtive), cumple con la ventaja de ser de Código Abierto y de distribución libre, sin límites en la cantidad de bases de datos que pueda manipular.

Sin embargo, en la industria de las Tecnologías de la Información, existen otros manejadores de base de datos que también son de Código Abierto y de distribución libre, como es el caso de PostgreSQL, JavaDB y Apache Derby, de la misma forma existen manejadores de base de datos como Oracle y DB2 que cuentan con versiones de distribución libre, pero no son de código abierto, estas versiones están limitadas en capacidad o escalabilidad.

Cualquiera de estos manejadores de bases de datos tienen a SQL, Structured Query Lenguaje o Lenguaje de Consulta Estructurado, como lenguaje compatible entre ellos, sin embargo existen sentencias propias de cada manejador y el cambiar entre uno y otro puede agregar un grado de dificultad considerable.

De la misma forma en que se instaló Apache y PHP en los Servidores de Aplicativos, la instalación de MySQL mediante el instalador de Webstack se realiza de manera sencilla, mediante los comandos **./install amp**, el cual hace la copia de los archivos ejecutables o binarios y configura los servicios de MySQL. Posterior a la instalación, es necesario inicializar el servicio de la misma forma en que se hizo con el Servidor de Aplicativos, mediante el comando **svcadm enable sun-mysql50** y el comando **svcs | grep mysql** para verificar que el servicio está inicializado (Figura 4.26).

```

192.9.198.190 - PuTTY
colaboracion-db /export/software # zonename
colaboracion-db
colaboracion-db /export/software # ./install amp
sun-mysql50.pkg...

WS104 & browser could not be invoked to display the Product Registration page. You may register the product by opening the fo
llowing URL in a browser:
file:///tmp/webstack_register.root.html

Installation was successful.
colaboracion-db /export/software # svcadm enable sun-mysql50
colaboracion-db /export/software # svcs | grep mysql
online          13:56:48 svc:/application/database/sun-mysql50:default
colaboracion-db /export/software # █
    
```

Figura 4.26 Habilitar el servicio de MySQL

La instalación del Webstack en cada uno de los Contenedores se realiza de la misma forma, dependiendo del propósito del Contenedor es el software que se instala, Apache y PHP en los servidores Web y de Aplicación y MySQL en los servidores de Base de Datos. Al finalizar la instalación en todos los contenedores, los elementos instalados se muestran en la figura 4.27.

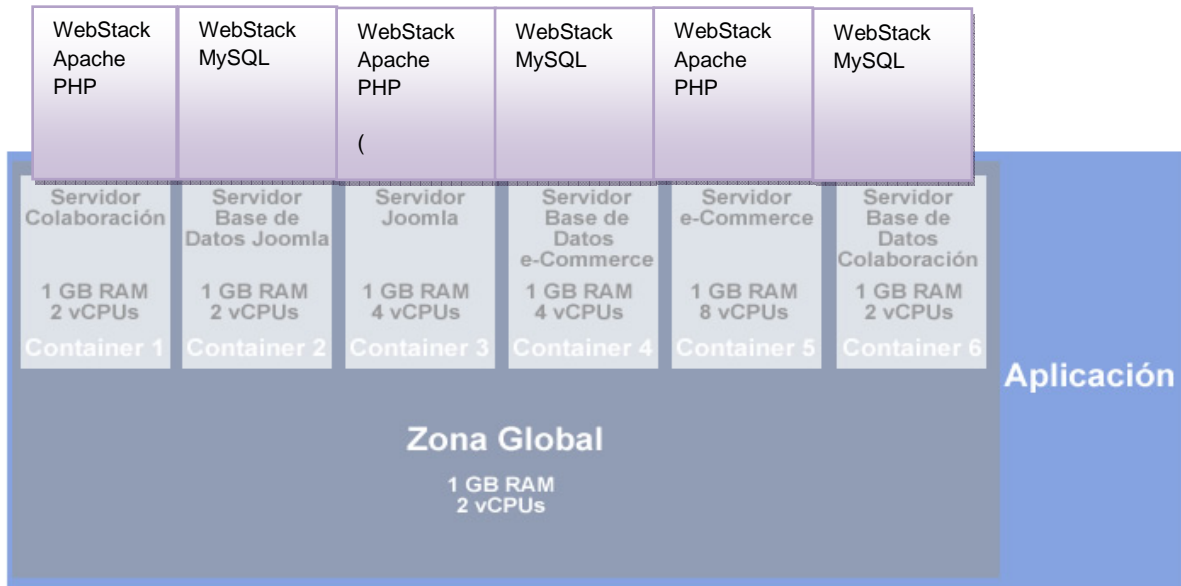


Figura 4.27 Instalación de Aplicaciones en los Solaris Containers

### 4.5.3. Configuración Webstack

Webstack fue configurado de acuerdo a cada una de las aplicaciones que en él residen, estas configuraciones incluyen desde creación de usuarios, inicialización de bases de datos y copia de archivos de las aplicaciones, culmina con la instalación y configuración de las aplicaciones.

El orden de configuración, preferentemente es Servidor de Base de Datos primero y luego el Servidor de Aplicaciones, esto debido a que es necesario tener un acceso y base de datos lista para instalar los aplicativos.

A continuación se muestra la configuración del Servidor de Base de Datos, en el proceso se revisan parámetros de red, estado del manejador de base de datos y posteriormente una configuración dentro del manejador de base de datos, la cual consiste en generar un usuario y su acceso a una base de datos específica.

En el Contenedor que corresponda al Servidor de base de datos, es necesario acceder a la consola de MySQL (Figura 4.28).

```

colaboracion-db / # /opt/webstack/mysql/bin/mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
    
```

Figura 4.28 Conexión al servidor de base de datos

Posteriormente se genera el usuario para la base de datos, los permisos de acceso y finalmente la base de datos es creada. A manera de estandarizar, la base de datos tiene el mismo nombre que el usuario creado, otorgándole todos los permisos de modificar toda la base de datos (Figura 4.29).

```

colaboracion-db / # /opt/webstack/mysql/bin/mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE USER 'colaboracion-db'@'192.9.198.185' IDENTIFIED BY '****';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT USAGE ON * . * TO 'colaboracion-db'@'192.9.198.185' IDENTIFIED BY '****' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE DATABASE IF NOT EXISTS 'colaboracion-db' ;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| colaboracion-db |
| mysql |
| test |
+-----+
4 rows in set (0.01 sec)

mysql> exit;
Bye
colaboracion-db / #
    
```

Figura 4.29 Configuración del servidor de base de datos

Con esto se finaliza la configuración del Servidor de Base de Datos. Seguido de este paso, es necesario configurar el aplicativo.

Primero se copian los archivos fuentes del aplicativo al directorio de http en Apache, el cual tiene una dirección específica en Webstack: `/var/opt/webstack/apache2/2.2/htdocs/`, todo lo que se encuentre dentro de este directorio será publicado mediante Apache Web Server en respuesta a las peticiones que se hagan desde un navegador de internet. La figura 4.30 muestra como se hace la copia de los archivos desde un cliente de transferencias de archivos llamado FileZilla.



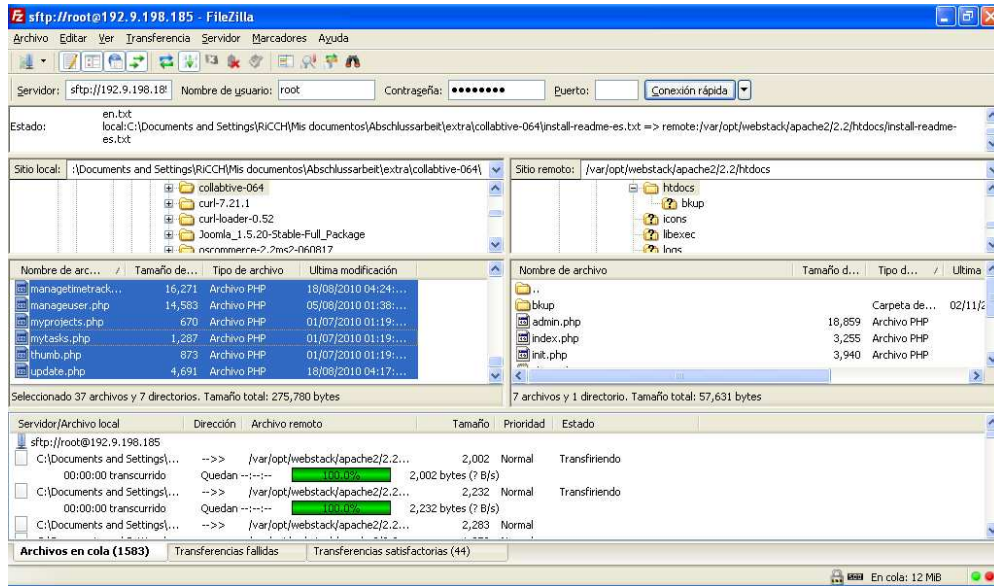


Figura 4.30 Transferencia de archivos

Es necesario hacer una verificación de los archivos y su integridad directamente en el Contenedor.

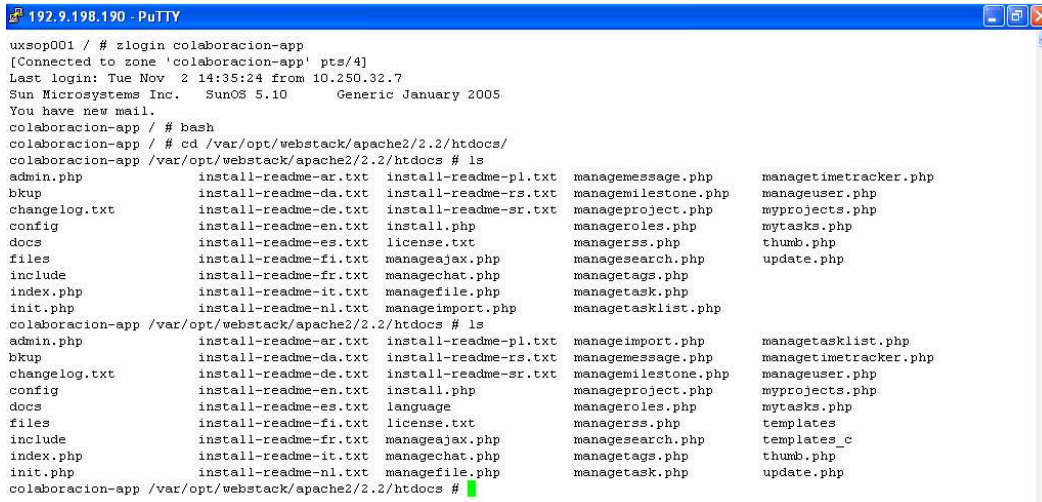


Figura 4.31 Integridad de archivos transferidos

En el caso particular del Servidor del Sitio de Colaboración, al momento de acceder mediante un navegador de internet, la siguiente página es la que se muestra:

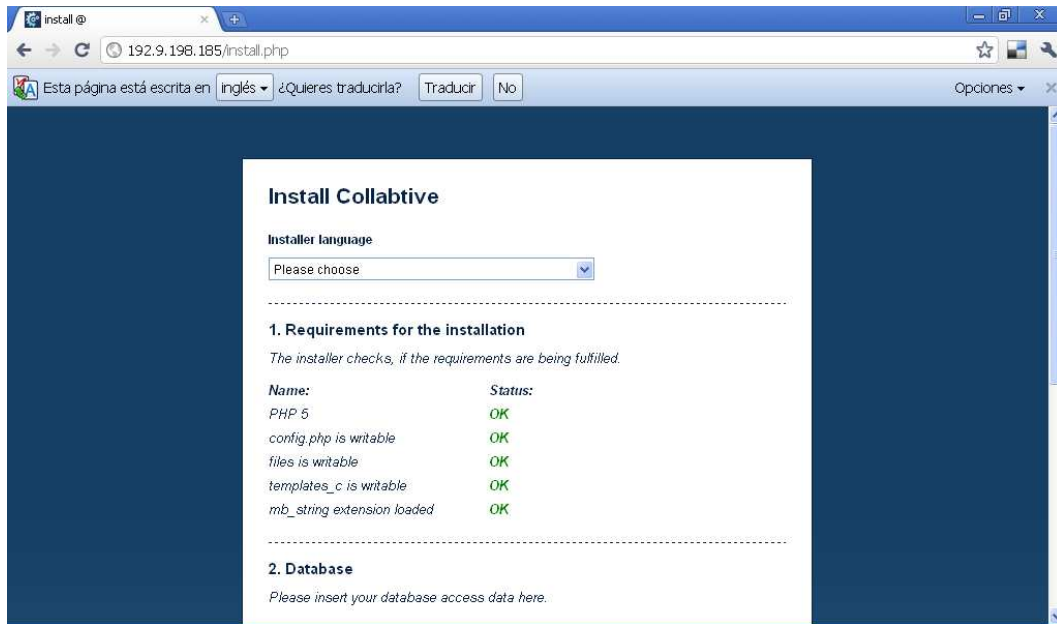


Figura 4.32 Pagina del Sitio de Colaboración

Esto nos indica que el servidor de aplicaciones está ejecutándose y respondiendo de manera adecuada. El proceso de instalación del aplicativo es único en cada caso, pero tiene puntos en común, la configuración de la base de datos requiere los parámetros de dirección IP, nombre de usuario y contraseña, tal cual se hizo en la configuración del servidor de base de datos.

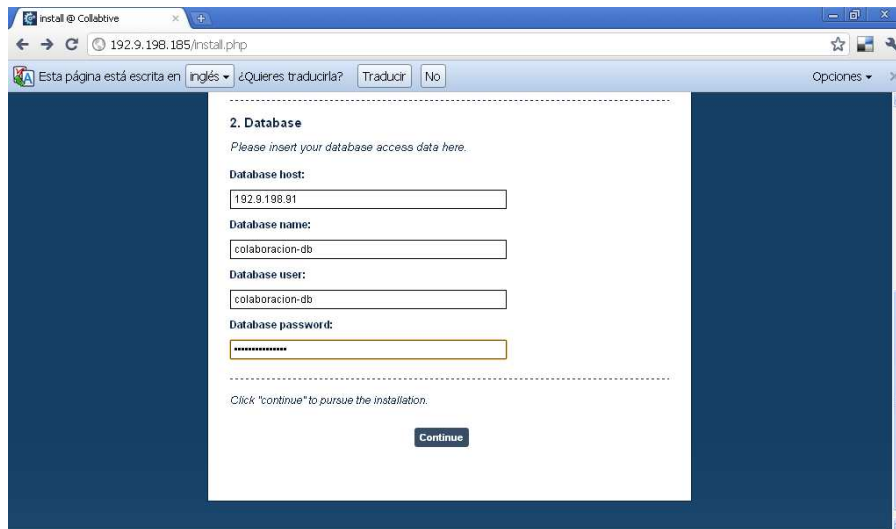


Figura 4.33 Configuración de la base de datos

# Capítulo 5

---

*En este capítulo se tomará como base la metodología general de proyectos descrita en los capítulos anteriores y se evaluará la implementación mediante la realización de pruebas.*

## 5. EVALUACIÓN

La metodología antes descrita (Iniciativa de negocio, diagnóstico, diseño, implementación y evaluación) ayudará a concluir, si la técnica cumple con los alcances esperados, como lo son la maximización de recursos, ahorro en el consumo de energía, ahorro en espacio, entre otros. Con la finalidad de evaluar la implementación de la virtualización que se usó en el presente trabajo de Tesis, se partió de una base, es decir, algo que, como se verá en los siguientes subtemas, sirve como referencia para confirmar al final, si la técnica cumplió con los resultados esperados o bien, si no cumplió y poder hacer una comparación entre el “antes de virtualizar” y el “después de virtualizar”.

Los datos comparados están basados en parámetros de evaluación, definidos además con criterios de aceptación, es decir una breve explicación sobre qué tipo de cambio antes y después de virtualizar es el conveniente.

Como resultado, se espera obtener datos que demuestren que la técnica de virtualización de servidores usando Solaris Containers es óptima y que en efecto, los consumos de energía disminuyen, así como el ahorro en espacio, la administración se simplifica y los recursos del servidor se aprovechan al máximo.

### 5.1. Definición de parámetros

En la figura 5.1 se especifican los parámetros de evaluación que son características con métricas que ayuden a las distintas áreas involucradas a tener información útil y tomar acciones sobre el total del sistema informático, optimizando con ello aspectos como los comentados en el tema 1, presupuestos para los interesados en la parte financiera, mejor aprovechamiento para los administradores de sistemas y un correcto funcionamiento para los usuarios.

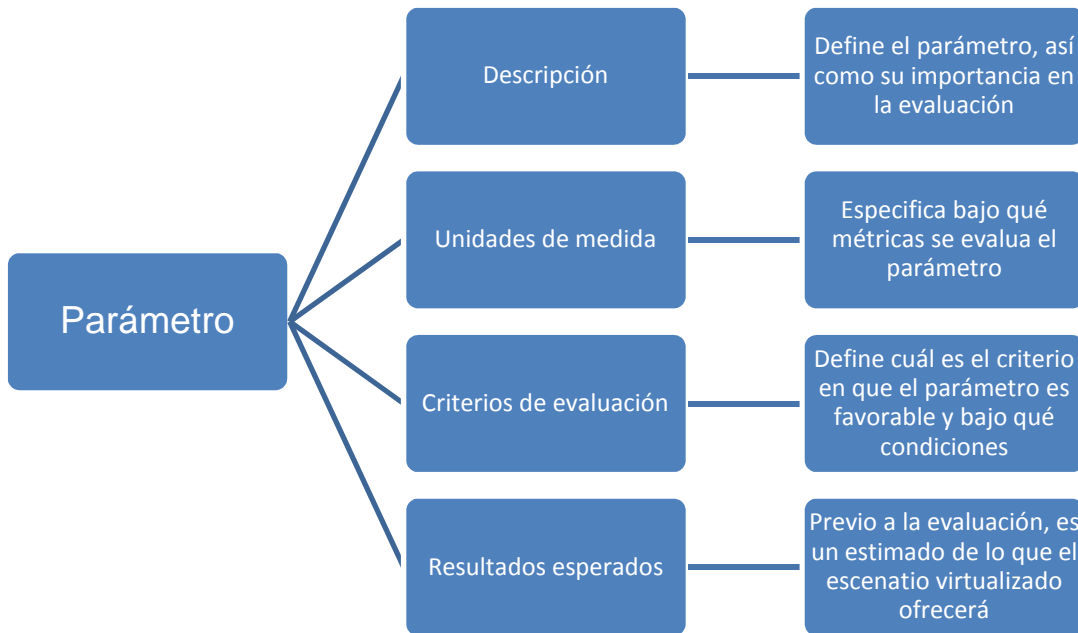


Figura 5.1 Parámetros de evaluación

- **Parámetro de evaluación:** Porcentaje de utilización de recursos
  - **Descripción:** Se define como recursos de un servidor al CPU y a la memoria disponible.
  - **Unidades de Medida:** El procesador o CPU se mide en % de ciclos de reloj usados en procesos y % de ciclos de reloj ociosos. La memoria se mide en % de uso en referencia al 100% que es la memoria instalada físicamente. Los sistemas operativos modernos, cuentan con procesos que monitorean los recursos del servidor, facilitan la medición de estos y proveen información extra, como el uso de recursos por cada proceso en el sistema.
  - **Criterio de evaluación:** Como parte de los activos de una organización, los recursos de un servidor deben ser optimizados, con el fin de obtener un mejor Retorno de Inversión, a mayor % de utilización sin disminuir el nivel de servicio, mejor se aprovechan dichos recursos. Organizaciones buscan que un servidor tenga al menos 30% de uso de recursos en promedio.
  - **Resultados esperados:** Un servidor virtualizado, al consolidar diversos servicios, utiliza mayor cantidad de recursos de cómputo, se espera que, a diferencia de los servidores no virtualizados, la propuesta hecha en este trabajo alcance un 20% de uso de recursos en promedio.
  
- **Parámetro de evaluación:** Tiempo de recuperación.
  - **Descripción:** Al ofrecer un servicio, las organizaciones buscan tener un mayor tiempo de disponibilidad, es decir, del tiempo que se necesite ofrecer un servicio, la mayor parte de ese tiempo esté disponible para ser usado. Para ofrecer una mayor disponibilidad, se necesita tener un menor tiempo de recuperación debido a que se calcula con la siguiente fórmula:

$$\text{Disponibilidad} = \text{MTTF}/(\text{MTTF}+\text{MTTR})$$

Donde

MTTF: Tiempo promedio para fallar

MTTR: Tiempo promedio de recuperación.

La disponibilidad contempla varias situaciones de falla o de mantenimiento, sin embargo para la evaluación en este trabajo se asume que todos los componentes funcionan correctamente y únicamente se detienen los servicios, se apaga el servidor virtual y se reinicia hasta su correcto funcionamiento. Es por ello que se decidió usar este parámetro y no el de disponibilidad.

- **Unidades de Medida:** El parámetro de tiempo de recuperación está en unidades de tiempo.
- **Criterio de evaluación:** Una mayor disponibilidad siempre es buscada por el área operativa de la organización, por ello se busca un menor tiempo de recuperación.
- **Resultados esperados:** Como ya se mencionó, en este trabajo se evaluó únicamente el tiempo que tarda en detener los servicios y reiniciar el sistema operativo, hasta que los servicios están nuevamente disponibles con toda la funcionalidad. Se espera que el tiempo de recuperación en un escenario virtualizado se reduzca hasta un 30% en referencia a un escenario no virtualizado.
- **Parámetro de evaluación:** Espacio físico
  - **Descripción:** Es el espacio vertical que ocupa un servidor en un Gabinete de Montaje o *Rack*. Dichos gabinetes y los dispositivos que en ellos se montan están alineados al estándar EIA-310.
  - **Unidades de Medida:** El estándar EIA-310 establece a las llamadas Unidades de Rack o Rack Units, equivalentes a 1.75 pulgadas o 44.45 mm, como medida de separación vertical, en la que se encuentran orificios para fijar los dispositivos al gabinete.
  - **Criterio de evaluación:** Optimizar el espacio, quiere decir tener mayor densidad de cómputo en un menor espacio. Muchas veces las organizaciones rentan dichos espacios, por lo que puede significar en un ahorro en el presupuesto.
  - **Resultados esperados:** Comparando un sistema virtualizado con un no virtualizado, se espera que el primero tenga una mayor densidad de cómputo y la mayor cantidad de aplicaciones ejecutándose en él con el menor espacio posible. Esperamos reducirlo al menos a la mitad en relación al escenario de servidores no virtualizados.

- **Parámetro de evaluación:** Energía eléctrica consumida.
  - **Descripción:** Se refiere a la cantidad de energía que consume el dispositivo electrónico.
  - **Unidades de Medida:** Típicamente para cualquier dispositivo eléctrico, electromecánico y electrónico la medida para calcular la energía eléctrica consumida es el Watt/Hora. Por practicidad, la métrica en este trabajo se hará con el consumo eléctrico indicado por el fabricante.
  - **Criterio de evaluación:** Debido a que la energía eléctrica es un servicio que se paga, optimizar el consumo de energía, se da cuando el servicio se presta con menor cantidad de energía eléctrica consumida.
  - **Resultados esperados:** El sistema virtualizado tiende a consumir menor cantidad energética, al consolidar servicios en una menor cantidad de servidores. Esperamos que al final de la evaluación, los servidores virtualizados disminuyan el consumo eléctrico al menos a la mitad, tomando como referencia su equivalente en servidores no virtualizados.
  
- **Parámetro de evaluación:** Energía térmica consumida.
  - **Descripción:** Se refiere a la cantidad de energía térmica que consume el dispositivo electrónico.
  - **Unidades de Medida:** El calor es energía y generalmente se expresa en Jules, BTU, toneladas o calorías. Las mediciones habituales de tasa de energía térmica producida para equipos son BTU por hora, toneladas por día y Jules por segundo. La tendencia internacional entre las organizaciones y los estándares indican que todas las mediciones de capacidad energética y de refrigeración deben estar en Watts.
  - **Criterio de evaluación:** Los equipos eléctricos que se encuentran en la sala del centro de datos o la red se refrigeran mediante aire, para reducir el calor que producen y mantener su temperatura en condiciones aceptables según sus especificaciones técnicas.
  - **Resultados esperados:** El sistema virtualizado tiende a consumir menor cantidad energética, al consolidar servicios en una menor cantidad de servidores. Esperamos que al final de la evaluación, los servidores virtualizados disminuyan el consumo térmico al menos a la mitad, tomando como referencia su equivalente en servidores no virtualizados.

- **Parámetro de evaluación:** Costo total de propiedad o *Total Cost of Ownership* (TCO).
  - **Descripción:** El costo total de propiedad, es la suma de todos los costos directos e indirectos de poseer un bien, esto incluye el costo de adquisición y los costos asociados a la operación y al mantenimiento, de la misma forma es considerado el costo asociado por tener un servicio fuera de línea.
  - **Unidades de Medida:** Este aspecto financiero es uno de los mayores impulsores para la virtualización, un estudio de especialistas en esa área determina el monto estimado que representa el poseer un bien, la unidad de medida es el monto monetario.
  - **Criterio de evaluación:** El estudio financiero del costo total de propiedad considera la mayor cantidad de aspectos relacionados con el bien poseído, entre menor sea este costo, mejor será para la organización. Para este trabajo de tesis, no profundizaremos en este parámetro, debido a que estará basado en experiencias e información pública.
  - **Resultados esperados:** Un escenario de servidores virtualizados debe representar un menor costo total de propiedad. Para un ejercicio de evaluación, se consideró el costo de operación, costo de licencias y costo de soporte por parte del proveedor como un porcentaje del costo de adquisición, esto en una proyección a 3 años, tiempo promedio mínimo de vida de un servidor.

En los siguientes subtemas se presenta la recopilación de los datos a evaluar así como su interpretación y por último definiremos si el escenario virtualizado cumplió con los resultados esperados.

## 5.2. Metodología de Evaluación

### 5.2.1 Porcentaje de utilización de recursos

El monitoreo de la utilización de recursos en el servidor se realiza con comandos nativos del sistema unix que proporcionan información para la determinación del estado del servidor en cuanto al aprovechamiento recursos.



Estos comandos suelen ser complicados de interpretar si se desconoce su sintaxis. Existen herramientas de monitoreo de recursos más amigables en su uso y que hacen una mejor presentación de resultados para facilitar la interpretación. Sin embargo, suelen ser licenciadas y su ejecución puede implicar cierto consumo de recursos para recolectar la información. Los principales comandos utilizados para el monitoreo de los recursos del servidor se describen a continuación:

- **sar** (System Activity Reporter)

Este comando obtiene datos del servidor sobre utilización de CPU, utilización de memoria, reporta el estado de los discos, llamadas de procesos al sistema, entre otros.

Uso: sar (opciones) (intervalo de medición) (número de mediciones)

Opciones:

-sar -u 1 5 Obtiene 5 muestras sobre la utilización de CPU con un intervalo de 1 segundo. La columna %idle nos presenta el porcentaje de disponibilidad del servidor al momento de la muestra y %wio se refiere al porcentaje de procesos encolados esperando poder hacer uso de los dispositivos.

SunOS uxsop001 5.10 Generic_142909-17 sun4v					
	%usr	%sys	%wio	%idle	
22:52:24	0	1	0	99	
22:52:25	0	1	0	99	
22:52:26	0	1	0	99	
22:52:28	1	1	0	98	
22:52:29	0	2	0	98	
22:52:30	0	2	0	98	
Average	0	1	0	99	

-sar -q 1 5 Muestra información sobre procesos encolados.

-sar -d 1 5 Muestra información sobre la actividad en los discos.

-sar -w 1 5 Muestra información sobre el estado de la memoria swap

-sar -g 1 5 Muestra información de memoria disponible y memoria swap.

Para consultar todas las opciones del comando sar basta con consultar el manual disponible al ejecutar el comando man sar. El comando sar es utilizado de dos maneras para obtener muestras indicando un intervalo y el número de muestras a obtener como se

describió anteriormente, la segunda manera, es utilizarlo para extraer información previamente recolectada en la ruta */var/adm/sa*. La información almacenada en esta ruta se realiza agregando dos entradas al cron del usuario root.

```

uxsop001 / # crontab -l
#ident "@(#)root 1.21 04/03/23 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] &&
/usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ____slave_kdcs____
0 2 2 * * /opt/SUNWexplo/bin/explorer -q # SUNWexplo
# Entrada para recolectar informacion SAR
0,5,10,15,20,25,30,35,40,45,50,55 0-23 * * 0-6 /usr/lib/sa/sa1
59 23 * * 0-6 /usr/lib/sa/sa2 -s 0:00 -e 23:55 -i 600 -A
    
```

La primer entrada */usr/lib/sa/sa1* es un script que utiliza *sar* para recolectar información sobre el desempeño del servidor en un archivo binario, asegurándose de que cada día tenga su propio archivo.

La segunda entrada */usr/lib/sa/sa2* es el script que se encarga de realizar la conversión del archivo binario creado por día y depurar los archivos con mas de 7 días de antigüedad.

- **vmstat** (Virtual Memory Statistic)

Reporta estadísticas que mantiene el kernel sobre los procesos, la memoria y otros recursos del sistema.

Uso:

*vmstat* (opciones) (intervalo de medición) (número de mediciones)

kthr			memory				page				disk				faults			cpu			
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	m0	m1	m4	m1	in	sy	cs	us	sy	id
30	0	0	21311704	5881448	4	18	3	0	0	0	1	1	11	0	1	573	527	519	0	0	99
0	0	0	21227024	5786872	0	17	0	0	0	0	0	0	0	0	0	587	1141	601	0	0	100
0	0	0	21226768	5786616	0	0	0	0	0	0	0	0	0	0	0	606	717	341	0	0	100
0	0	0	21226768	5786616	0	0	0	0	0	0	0	0	0	0	0	537	704	302	0	0	100
0	0	0	21226768	5786624	2	0	0	0	0	0	0	0	0	0	0	594	802	339	0	0	100

La primera línea del resultado presenta un promedio de estos valores del sistema desde el ultimo reinicio del sistema. Del primer bloque *kthr* relacionado con los procesos,

la columna *w* suma los procesos que han sido llevados al área de swap en los últimos segundos por lo que deberá permanecer este valor siempre en 0.

El segundo bloque **memory** muestra los recursos disponibles, en kilobytes, en cuanto a memoria swap y memoria física.

El tercer bloque **disk** reporta información del número de operación por segundo realizados en discos del sistema operativo.

Los bloques restantes son para información sobre procesos ejecutándose en el CPU, así como su desempeño.

- **iostat** (Input/Output statistics)

Al igual que los comandos anteriores este comando funciona para obtener estadísticas sobre la actividad en los dispositivos de entrada y salida, así como reportar también información de CPU. Sin embargo, lo utilizamos principalmente para obtener datos sobre la utilización de los discos.

Uso:

`iostat (opciones) (intervalo de medición) (número de mediciones)`

```

uxsop001 / # iostat 1 10

          tty          md0          md1          md4          md10          cpu
      tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
0         2   5   1  33    0  0   9   0  0  16   4   1  16   0  1  0 99
0        48  14   3  18    0  0   0   0  0   0  14   3  11   5  4  0 92
0        19   0   0   0    0  0   0   0  0   0   0   0   0   3  5  0 92
0        16   2   1   7    0  0   0   0  0   0   2   1   5   4  4  0 92
0        20   0   0   0    0  0   0   0  0   0   0   0   0   5  4  0 92
    
```

Como se puede observar por cada dispositivo como terminal (tty) o discos (md0, md1, md4, md10) se presenta información sobre kilobytes escritos por segundo, transferencias por segundo y tareas que están realizando petición a ese dispositivo.

**5.2.2. Costo total de propiedad o Total Cost of Ownership (TCO)**

Como ya se ha mencionado a lo largo de esta Tesis, el Coste Total de Propiedad (TCO en sus siglas en inglés) es un tema muy importante de negocios, ya que una vez entendido se vuelve un factor crítico para identificar el Retorno de Inversión (ROI por sus siglas en inglés) de un proyecto de TI.

Cuando una organización o una compañía está desarrollando un proyecto, siempre se toma en cuenta el valor del TCO cuando se empieza a tocar el delicado tema del gasto que representará para la compañía, el llevar a cabo un proyecto. Para poder medir el éxito de un proyecto se debe de medir el TCO antes de llevar a cabo el proyecto y después de un tiempo ya implementado, para poder apreciar el coste real que representó el proyecto.

Medir el TCO no es fácil y la mayoría de las veces depende de los valores que realmente le interesen a la compañía, es decir, una organización o compañía debe de preguntarse cuáles valores son importantes para ella, como pueden ser el ahorro de energía, el costo de espacio, el costo de administración de los equipos, etc.

Existen varios modelos de TCO, entre los cuales se encuentran:

- Coste Total de Propiedad (TCO). Un concepto desarrollado por Gartner, es una metodología que se enfoca en los costes del entorno de TI (espacio, energía, recursos, etc).
- Impacto Económico Total (TEI). Desarrollado por Giga Information Group, se enfoca en cómo una solución impacta todas las áreas de negocio de una organización.
- Valor Total de Propiedad (TVO). Desarrollado por Lucidus Ltd. Esta metodología tiene como principal objetivo, no solamente medir el impacto de un proyecto, sino también medir el valor agregado que aporta el proyecto a la organización.

Para este trabajo de Tesis, nos basamos en el modelo de Gartner para medir el TCO que representa el proyecto.

Gartner define el Coste Total de Propiedad (en adelante TCO), como todo el coste asociado a un proyecto de TI, es decir, no únicamente toma en cuenta el coste del *hardware*, también toma en cuenta otros costes que van ligados a éste, como son: el costo de adquisición del *hardware*, la instalación, la implementación, la administración, el mantenimiento e incluso la desinstalación del *hardware*. Normalmente, el costo de adquisición del hardware representa un 15% del TCO y el resto corresponde al costo de administración e implementación del proyecto en su totalidad.

El modelo de Gartner define 2 categorías para definir costo: Costos Directos y Costos Indirectos.

Dentro de la primera categoría, se encuentran el costo del *hardware* y del *software*, administración de sistemas, almacenamiento, soporte, recursos, servicios de consultoría, etc.

Los Costos Indirectos abarcan la parte del servicio que ofrece la solución del proyecto de TI, como es la disponibilidad y el *downtime* en caso de falla del sistema.

Para realizar el análisis de TCO, existe una herramienta desarrollada por Gartner llamada *Gartner's TCO Manager*, la cual contiene una extensa base de datos recolectada de 300 centros de datos de 200 compañías diferentes. La herramienta tiene un costo de licencia, no existen versiones de demostración; pero la recomendamos ampliamente para aquellas organizaciones que cuenten con los recursos necesarios.

Se realizó un análisis de TCO basándonos en los siguientes supuestos:

- El siguiente análisis de TCO es un análisis simple de los ahorros que se generan al consolidar servidores mediante la virtualización.
- El inventario usado como base del análisis está compuesto por 3 servidores físicos, que son los que se tomarán en cuenta para la consolidación mediante virtualización.
- Los servidores del inventario incluye las marcas Sun Microsystems.
- El servidor Sun Fire T2000 es la plataforma usada para la virtualización. El costo del servidor y su configuración es de \$60,000 USD. Este costo es un estimado y no incluye descuentos.
- Todos los datos del análisis están basados en precios estimados. Los costos reales pueden diferir con los de este análisis.
- Los resultados obtenidos de este análisis representan un aproximado del ahorro que se puede lograr mediante la consolidación de servidores, pero no garantiza que los datos aquí obtenidos se cumplan.

Como primer paso, se obtiene el análisis de TCO de acuerdo a la arquitectura actual, es decir, antes de la virtualización de servidores. De esta manera después podremos hacer una comparativa de costos entre la situación actual y la propuesta.

Todos los datos contenidos en las tablas están expresados en dólares americanos y en tiempo corresponden a 1 año. Los rubros considerados se describen a continuación:

Hardware: Se refiere a los costos relacionados con la adquisición de Hardware

- Servidores: Es el costo de adquisición de los servidores, con todos los dispositivos internos que se han descrito en el capítulo 2 así como su instalación. Es un costo estimado basado en la información pública del fabricante.

- Consolas: Es el costo de adquisición de equipos de administración de los servidores. También es un costo estimado, basado en información pública del fabricante.
- Periféricos: Corresponde a los dispositivos adicionales y externos en los servidores o en las consolas.
- Componentes de Red: Se refiere a los componentes activos que permiten interconectar a los servidores, las consolas y otros dispositivos. Se consideran Switches, Ruteadores y Firewalls.
- Costo de Mantenimiento: Es un costo que el fabricante cobra como complemento de la garantía, ofreciendo servicios de reemplazo, atención telefónica y acceso a contenido remoto.
- Energía (Enfriamiento y Alimentación): Está basado en el costo aproximado del KW/h proporcionado por Comisión Federal de Electricidad, tomando en consideración los consumos oficiales establecidos en la documentación del Servidor. Así mismo, contempla el equivalente en KW necesarios para enfriar los servidores.

#### Software

- Licencias: Es el costo de las licencias de los programas utilizados en la implementación, contempla Sistema Operativo, agentes, Software de configuración y Aplicativos. Para este trabajo se usó Software de distribución libre, es decir, sin costo por descargarlo, instalarlo y usarlo.
- Costo de Mantenimiento: Es el costo asociado al mantenimiento del Software. Sin embargo, de la misma forma que con las licencias, el software que se usó en este trabajo, no contempla un costo de mantenimiento.

#### Administración

- Red: Se considera el costo asociado a un recurso para la administración de la Red, con ello se cubre todos los aspectos relacionados con la interconexión de los equipos.
- Servidores: Se considera el costo asociado a un recurso para la administración de los servidores, con ello se cubren los aspectos relacionados con Sistema Operativo y con Hardware.

- Almacenamiento: Se considera el costo asociado a un recurso para la administración del almacenamiento.
- Bases de Datos: Se considera el costo asociado a un recurso para la administración de las bases de datos, comúnmente es conocido como DBA, por sus siglas en inglés. Para este trabajo, es quien administra, configura y da mantenimiento a MySQL.
- Servidores de Aplicaciones: Se considera el costo asociado a un recurso para la administración lógica de los servidores de aplicaciones, para el caso de este trabajo, es quien administra, configura y da mantenimiento de Sun Webstack, en particular de Apache y PHP.

### Soporte

- Personal de Soporte: Es personal especializado y con entrenamiento suficiente para resolver problemas asociados con fallas de servidores, típicamente están disponibles las 24 horas y resuelven aquello que el personal de Administración no resuelve.
- Entrenamiento: Es el costo asociado a cursos necesarios para operar los servidores, se consideran costos aproximados a los publicados por el fabricante.
- Adquisiciones: Considera cualquier otro bien adquirido, no relacionado directamente con Hardware, pero necesario para la operación y soporte de este. En el caso de este trabajo, no considera ninguna adquisición fuera de lo ya establecido en el Hardware.
- Viaticos: Gastos asociados con los viajes y visitas fuera del sitio donde se encuentran los servidores. En este trabajo no se considera ningún gasto de este tipo, ya que asume que no son necesarios.
- Contratos de Soporte: Es un acuerdo entre el fabricante y la organización que posee los servidores, en el cuál se establece un nivel de servicio, el cuál estipula actividades de soporte con un tiempo de respuesta definido, así como otras actividades que incluyen reconfiguración, actualizaciones y mantenimientos.
- Actividades Extras: El costo estimado por cualquier actividad de soporte no contemplada en los puntos anteriores. En este trabajo no se considera ningún costo de este tipo.

### Implementación

- Desarrollo/Personalización/Integración: Es el costo asociado a la configuración de los servidores, basado en lo que está establecido en el capítulo 4 de Implementación, subtema Configuración del Sistema Operativo Base.
- Entrenamiento: Es el costo asociado al entrenamiento necesario para la implementación, que están fuera del entrenamiento para la parte operativa y de soporte. En este trabajo no se consideran este entrenamiento, debido a que la implementación del escenario virtualizado lo realizan mediante una consultoría.
- Consultoría: Es el costo asociado a recursos externos que realizan la parte de implementación especificada en el capítulo 4 de implementación, subtemas relacionados con los Contenedores. En el ejercicio de TCO para el escenario no virtualizado no se contempla, porque se refieren en particular a los Contenedores.
- Otros: Cualquier otro costo no especificado en la implementación, sin embargo en este trabajo no se contempla ninguno.

### Telecomunicaciones

- Red Local: Es el costo asociado a cableado dentro del centro de datos, abarca cableado UTP de Red y Fibras para el arreglo de disco.
- Red Externa: Es el costo asociado con la renta de enlaces externos mediante un Proveedor de Servicios de Internet, el cuál proporciona enlace externo.
- Acceso Remoto: Es el costo asociado con accesos remotos mediante una Red Virtual Privada, conocida como VPN por sus siglas en inglés. Este costo no se considera en este trabajo.

### Usuario Final

- Entrenamineto del Usuario Final: Es el costo asociado con el entrenamiento a personal que administra la aplicación, sin embargo no se consideran estos costos, debido a que el foco de este trabajo es la virtualización del Sistema Operativo.
- Entrenamiento informal del usuario: Es el costo asociado con el entrenamiento informal al usuario final, sin embargo este costo no se considera.



- Desarrollo en Casa de aplicaciones: Es el costo asociado con el desarrollo y adecuación de los aplicativos, sin embargo este costo no se considera, debido a que se utilizó un aplicativo con datos de ejemplo.
- Mantenimiento de Archivos: Es el costo asociado con el mantenimiento de archivos históricos, como son respaldos y manejo del ciclo de vida de la información, conocido en la industria como ILM por sus siglas en inglés.

#### Tiempo fuera de línea

- Tiempo fuera de línea planeado: Es el costo que se asocia a tener una aplicación fuera de línea de manera planificada, se contemplan las pérdidas económicas por falta de servicio.
- Tiempo fuera de línea no planeado: Es el costo que se asocia a tener una aplicación fuera de línea de manera no planificada, típicamente es debido a una falla. Al no ser planificada la puesta fuera de línea el costo es mayor en comparación a la planificada.

### 5.2.3. Espacio físico

Los racks son los armarios estandarizados que comúnmente se usan para instalar elementos electrónicos de forma ordenada. Estos armarios disponen de dimensiones perfectamente definidas y se fabrican mundialmente bajo la norma EIA 310-D. Se definió una unidad estándar de medida en cuanto a las dimensiones de los equipos y así facilitar el método de atornillarlos y sujetarlos de forma adecuada, esta medida se llama Unidad de Rack (Rack Unit) o comúnmente "U" .

Una unidad de rack "U" es la distancia que existe del centro de los barrenos separados por 12.7 mm. hasta el centro de los siguientes barrenos separados por 12.7 mm, esta unidad de rack corresponde con la medida americana habitual de 1,75" o bien 44.45 mm y se le denomina con 1U, esta medida describe la altura de los equipos montados sobre los racks de 19" o 23" de ancho.

Los racks cuentan con 2 tipos de esparcimiento, Universal y Ancho como se muestra en la figura 5.2, pero siempre se contabilizan por unidad de rack "U".

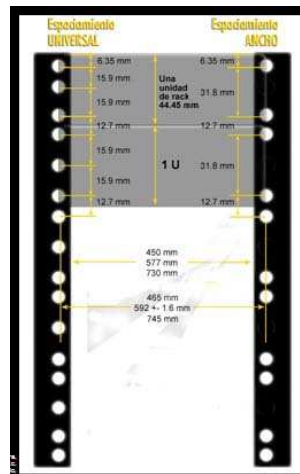


Figura 5.2 Estructura del rack

La metodología para evaluar el espacio físico, es de manera sencilla comparar el espacio físico vertical en Unidades de Rack entre el escenario virtualizado y el escenario no virtualizado, tomando como base el diagnóstico y el diseño hechos en el capítulo 3. Se toma la especificación de la ficha técnica que se ha elaborado en el capítulo 2. El espacio físico que utiliza un servidor se contabilizara por las Unidades de Rack “U” que ocupe por gabinete, no se tomar en cuenta el factor de la profundidad, que no está normalizada, y suele pasarse por alto.

#### 5.2.4. Energía eléctrica consumida

La consolidación física resultante de la virtualización reduce el consumo de energía de dos maneras, dado que se reduce la cantidad de servidores y también porque se elimina una parte de la energía que consumían los sistemas de enfriamiento.

La Comisión Federal de Electricidad (**CFE**) quien controla esencialmente todo el sector eléctrico, tiene diferentes tarifas de acuerdo al tipo de industria que solicita el servicio, lugar geográfico y tipo de tensión requerida. La Baja tensión son tensiones inferiores a 1 kV que se reducen todavía más para que se puedan emplearse en la industria, el alumbrado público y el hogar. Las tensiones más utilizadas en la industria son 220, 380 y 440 volt de corriente alterna y en los hogares entre 110 y 120 volt para la mayoría de los países de América y 220 volt para Europa.

En el proceso de evaluación del ahorro de energía se analizará y realizara una comparación entre el total del consumo de energía de los 6 servidores antes de la

virtualización, y el consumo energético después de la virtualización, cada servidor requiere de alimentación de energía en sus 2 fuentes para su funcionamiento, cabe mencionar que dichos servidores también requieren energía que para su enfriamiento.

La energía eléctrica se vende en unidades de energía denominadas kilovatios por hora (kW/h), que es la cantidad de energía despachada en una hora a un nivel de potencia de 1000 vatios (1 kW), con esta información de obtendrá el consumo anual de los servidores sin virtualizar y posteriormente virtualizados, así como el costo anual que representan estos consumos tomando como referencia el costo kW/h que define **CFE** para el Sector Publico en la ciudad de México.

En este caso se utiliza la tarifa de CFE para el sector público en la Ciudad de México para un consumo de baja tensión, ya que el servidor requiere de un voltaje de 100 a 200 V.

Se tiene que cada servidor tiene una potencia de 450 W, la potencia se define como la velocidad a la que se consume la energía, energía por unidad de tiempo (J/s) y ya que dichos servidores son de alta disponibilidad, significa que estarán encendidos las 24 horas los 365 días del año, con estos datos se obtendrá el consumo Anual en los 2 escenarios.

### 5.3. Obtención y Análisis de Resultados

#### 5.3.1 Porcentaje de uso de recursos

La obtención de resultados en el servidor se realizó de la siguiente manera:

- **Monitoreo del servidor**

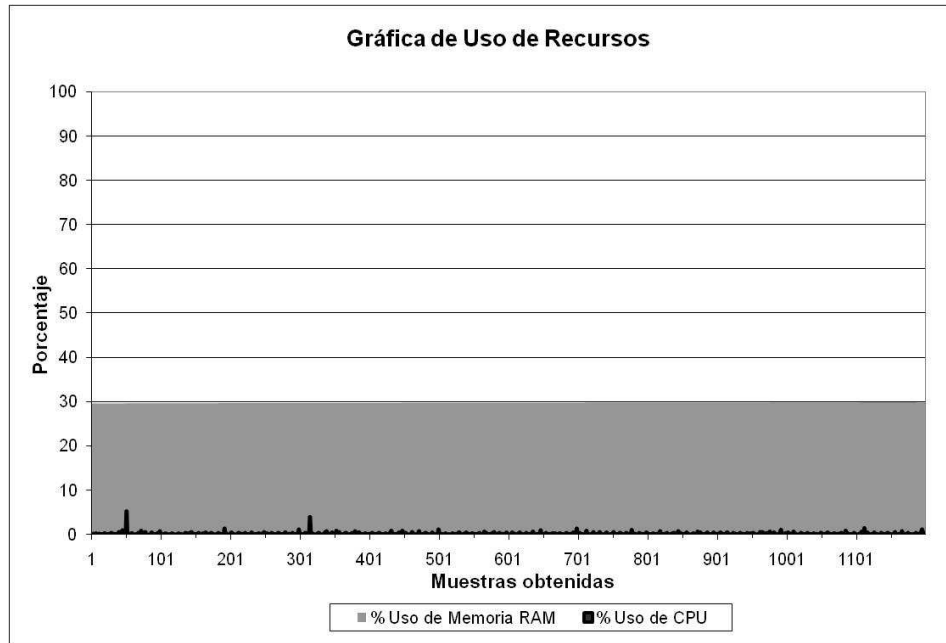
Después de instalado el Sistema Operativo Base, se obtuvieron estadísticas de uso de CPU, memoria y discos.

La tabla 5.1 muestra la información almacenada en la ruta **/var/adm/sa** del monitoreo implementado con el comando **sar**. Las muestras presentadas se obtuvieron cada hora durante un día completo, al final se encuentra el promedio de todas las muestras obtenidas.

Hora	%usr	%sys	%wio	%idle
00:00:01	0	0	0	100
01:00:01	0	0	0	100
02:00:01	0	0	0	100
03:00:01	0	0	0	100
04:05:00	0	0	0	100
05:00:01	0	0	0	100
06:00:01	0	0	0	100
07:00:01	0	0	0	100
08:00:01	0	0	0	100
09:00:02	0	0	0	100
10:00:01	0	0	0	100
11:00:01	0	0	0	100
12:00:01	0	0	0	100
13:00:01	0	0	0	100
14:00:01	0	0	0	100
15:05:00	0	0	0	100
16:00:01	0	0	0	100
17:00:01	0	0	0	100
18:00:01	0	0	0	100
19:00:01	0	0	0	100
20:00:01	0	0	0	100
21:00:01	0	0	0	100
22:00:01	0	0	0	100
23:00:01	0	0	0	100
23:50:01	0	0	0	100
Promedio	0	0	0	100

El servidor con el Sistema Operativo Solaris 10 se encuentra a un 100 % de disponibilidad en cuanto a uso de CPU, esto se debe a que solo ejecuta procesos mínimos para su funcionamiento. Al no estar brindando ningún servicio ni estar ejecutando procesos de aplicaciones o de base de datos, observamos que los recursos del servidor están siendo desperdiciados.

La grafica 5.1 muestra el resultado de los datos obtenidos de un monitoreo de 1200 muestras obtenidas, con el comando **top**.



Grafica 5.1 Uso de recursos

En la gráfica 5.1 se muestra que el consumo de memoria RAM se encuentra en un promedio de 29.77% equivalente a 2.34 GB de los 8 GB disponibles en el servidor. Mientras que el promedio de consumo de CPU es de 0.26%.

- **Monitoreo del servidor con aplicación sin Virtualización.**

**Prueba 1: Ejecutar la herramienta stress con la aplicación de Comercio Electrónico.**

**Objetivo:** Validar que el servidor sin virtualización con la aplicación de Comercio Electrónico es capaz de ejecutar los procesos esperados en tiempo y de forma exitosa.

**Realización de la Prueba y Resultados:**

Con ayuda de la herramienta stress se simularon 5000 procesos por hora haciendo uso de 128 Kb de memoria RAM y 5000 procesos utilizando los dispositivos de entrada y salida, estos son los procesos que se esperaran para la aplicación de Comercio Electrónico. Al ejecutar la herramienta se especifica el tiempo de duración.

```
uxsop001 / # stress --cpu 5000 --vm 5000 --vm-bytes 128K --timeout 1h --backoff 2 -i 5000
```

```
stress: info: [7508] dispatching hogs: 5000 cpu, 5000 io, 5000 vm,
0 hdd
stress: info: [7508] successful run completed in 3636s
```

Una vez concluida la ejecución se observa que todos los procesos indicados terminaron en el tiempo indicado, por lo que se demuestra que el servidor es capaz de soportar los procesos esperados en tiempo y forma.

## Prueba 2: Generar carga en la aplicación de Comercio Electrónico Sin Virtualización con la herramienta Apache JMeter.

**Objetivo:** Evaluar el uso de recursos de CPU y Memoria RAM en el Servidor sin Virtualizar.

### Realización de la prueba

Se configuro la herramienta Apache JMeter para generar 5000 peticiones para la aplicación de Comercio Electrónico con duración de 1 hora y se puede observar en la figura 5.3.

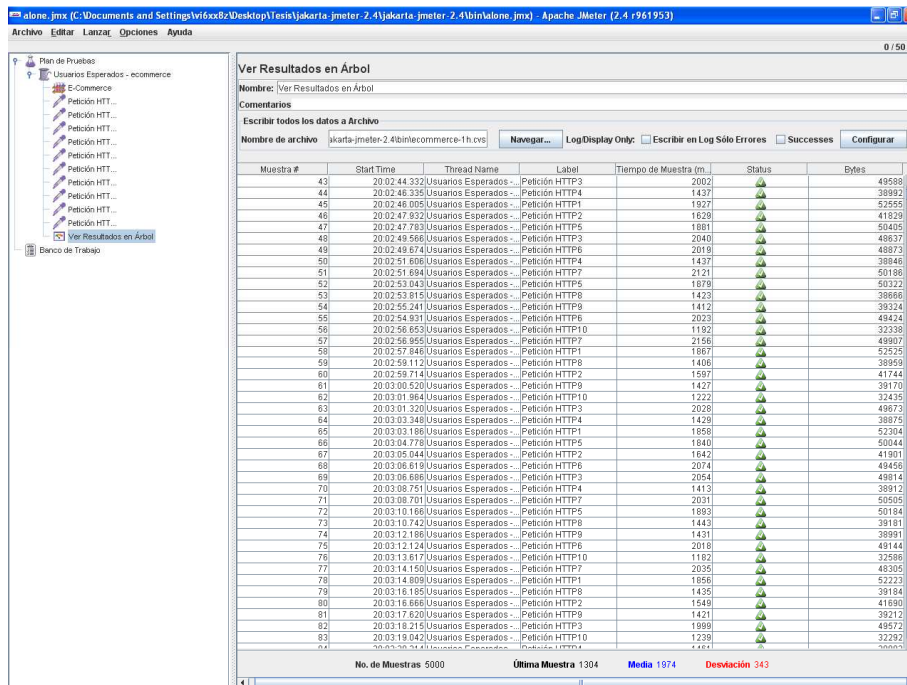
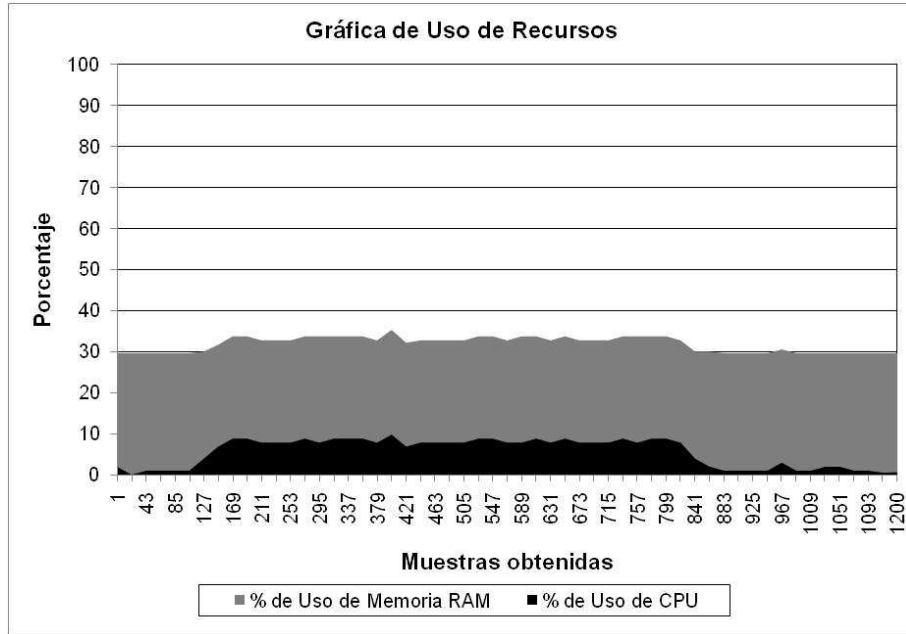


Figura 5.3 Apache JMeter simulación de carga de trabajo

Con los comandos sar y vmstat se obtuvieron los datos necesarios para la gráfica 5.2:

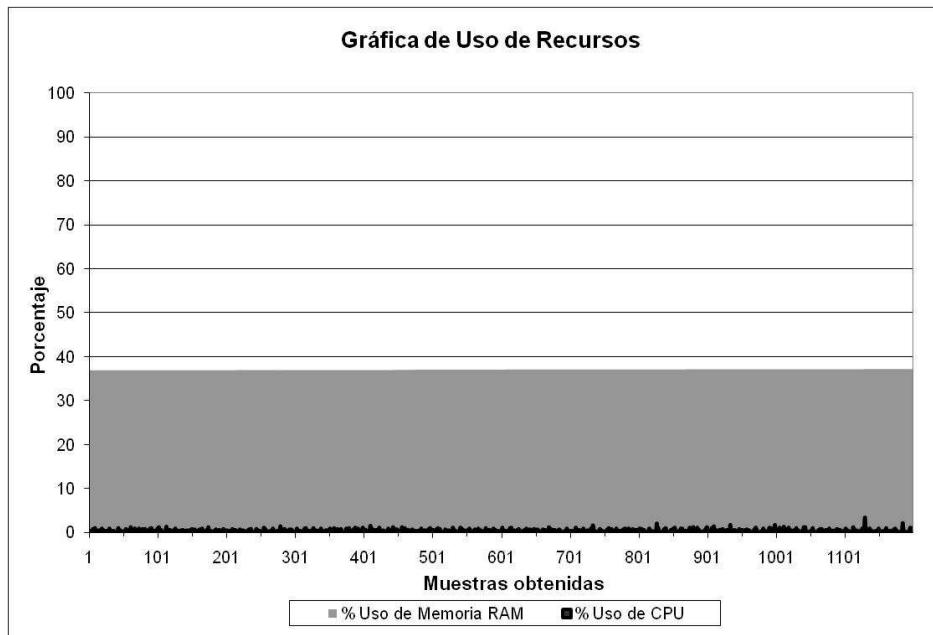


Gráfica 5.2

El promedio de uso de CPU es del 6.64% y de uso de Memoria RAM es de 31.84%.

- **Monitoreo del servidor con Solaris Containers**

Concluida la implementación de los Solaris Containers, se obtuvieron 1200 muestras para observar al consumo de los recursos del servidor. Los resultados del monitoreo se muestran en la gráfica 5.3.



Gráfica 5.3 Uso de recursos

El promedio del porcentaje de uso de recursos de CPU fue de 0.53 % y de uso de memoria RAM fue de 37.14 %, estos porcentajes se pueden observar en la gráfica 5.8.

- **Monitoreo del Servidor Virtualizado con Solaris Containers**

Una vez implementado el diseño de la arquitectura propuesta se realizaron las siguientes pruebas:

**Prueba 1: Ejecutar la herramienta stress en cada Solaris Container.**

**Objetivo:** Validar que el contenedor por aplicación es capaz de ejecutar los procesos esperados en tiempo y de forma exitosa.

**Realización de la Prueba y Resultados**

Se muestran los resultados arrojados por la herramienta por cada Solaris Container con su aplicación correspondiente.

```

colaboracion-app / # stress --cpu 200 --vm 200 --vm-bytes 512K --timeout 1h
--backoff 2 -i 200
stress: info: [17311] dispatching hogs: 200 cpu, 200 io, 200 vm, 0 hdd
stress: info: [17311] successful run completed in 3616s

colaboracion-db / # stress --cpu 200 --vm 200 --vm-bytes 512K --timeout 1h
--backoff 2 -i 200
stress: info: [17308] dispatching hogs: 200 cpu, 200 io, 200 vm, 0 hdd
stress: info: [17308] successful run completed in 3619s

ecommerce-app / # stress --cpu 5000 --vm 5000 --vm-bytes 128K --timeout 1h
--backoff 2 -i 5000
stress: info: [17309] dispatching hogs: 1500 cpu, 1500 io, 1500 vm, 0 hdd
stress: info: [17309] successful run completed in 3630s

ecommerce-db / # stress --cpu 5000 --vm 5000 --vm-bytes 128K --timeout 1h -
-backoff 2 -i 5000
stress: info: [17310] dispatching hogs: 5000 cpu, 5000 io, 5000 vm, 0 hdd
stress: info: [17310] successful run completed in 3634s

joomla-app / # stress --cpu 5000 --vm 5000 --vm-bytes 128K --timeout 1h --
backoff 2 -i 5000
stress: info: [17312] dispatching hogs: 5000 cpu, 5000 io, 5000 vm, 0 hdd
stress: info: [17312] successful run completed in 3636s

joomla-db / # stress --cpu 5000 --vm 5000 --vm-bytes 128K --timeout 1h --
backoff 2 -i 5000
stress: info: [17313] dispatching hogs: 5000 cpu, 5000 io, 5000 vm, 0 hdd
stress: info: [17313] successful run completed in 3638s

```



**Análisis:** Para las aplicaciones de Joomla y de ecommerce se generan 5000 peticiones por hora. En los servidores de aplicación, como ecommerce-app y joomla-app, y en los servidores de base de datos, como ecommerce-db y joomla-db, se realizó la misma prueba con las mismas características y el resultado fue exitoso, al generar una carga de 5000 procesor, cada Solaris Container cumple de forma eficaz los requerimientos para soportar la carga generada.

**Prueba 2: Generar carga hacia cada aplicación con la herramienta Apache JMeter.**

**Objetivo:** Evaluar el uso de recursos de CPU y Memoria RAM en el Sistema Operativo Base.

**Realización de la prueba**

Se configuro la herramienta Apache JMeter para generar 25000 peticiones para las aplicaciones de Comercio Electrónico (figura 5.4), Manejador de contenidos (figura 5.5) y Sitio de Colaboración (figura 5.6) con una duración de 5 horas, esto sería equivalente a 5000 usuarios por hora. Para el Sitio de Colaboración se configuraron 1000 usuarios con duración de 5 horas, equivalente a 200 usuarios por hora.

- Aplicación de Comercio Electrónico

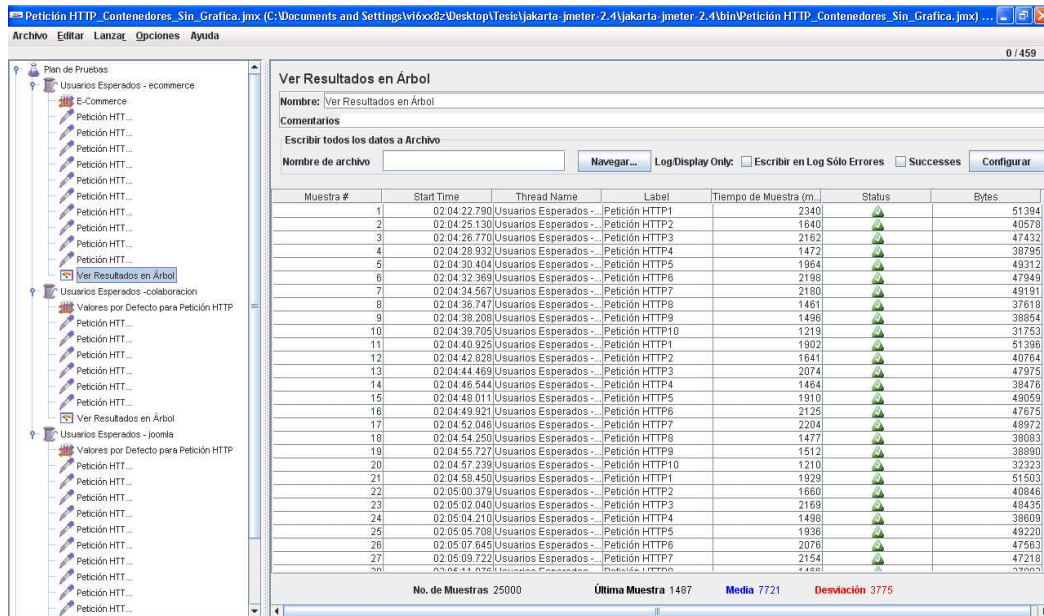


Figura 5.4 Apache JMeter simulación de carga de trabajo para Comercio Electrónico

- Aplicación de Manejador de Contenidos

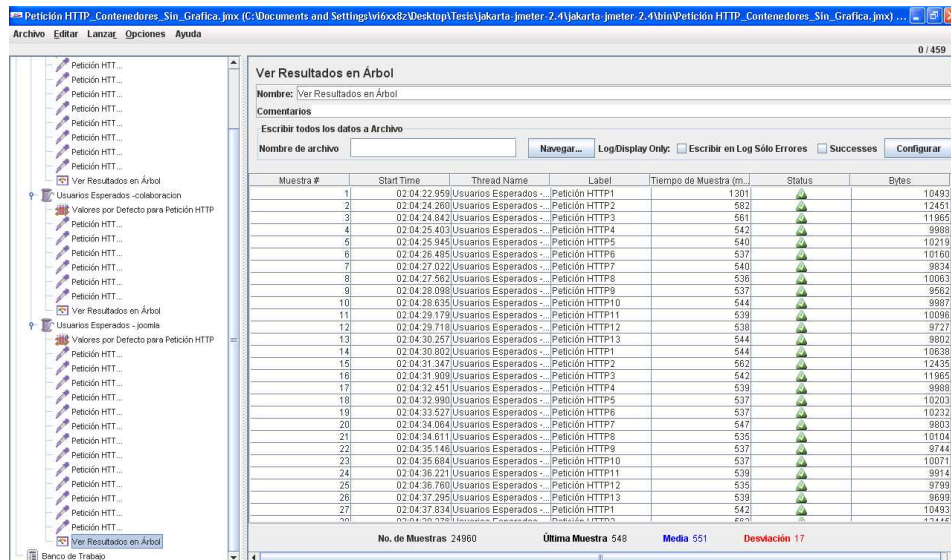


Figura 5.5 Apache JMeter simulación de carga de trabajo para Manejador de Contenidos

- Aplicación de Sitio de Colaboración

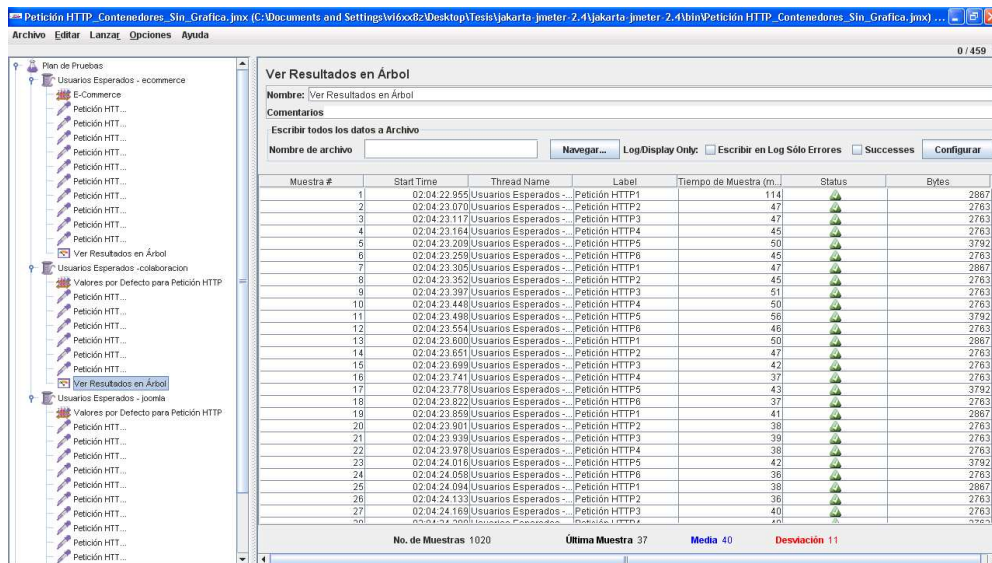
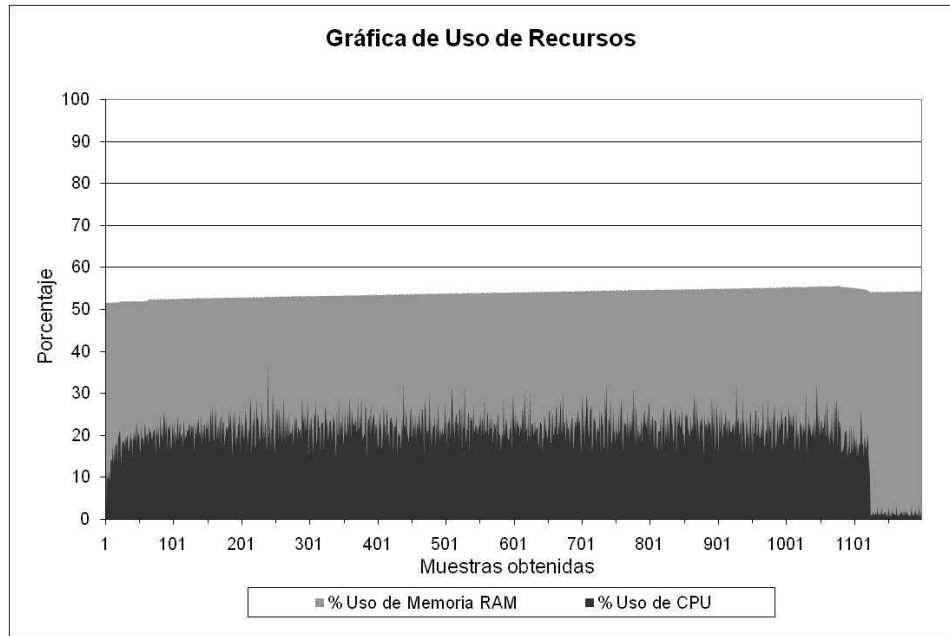


Figura 5.6 Apache JMeter simulación de carga de trabajo para Sitio de colaboración

Resultados

Del monitoreo implementado durante la ejecución de la prueba, con los comandos sar y vmstat se obtuvieron 1200 muestras para determinar el porcentaje de uso de CPU y el porcentaje de uso de Memoria RAM en el Sistema Operativo Base. El comportamiento de los recursos del servidor se muestra en la gráfica 5.4.

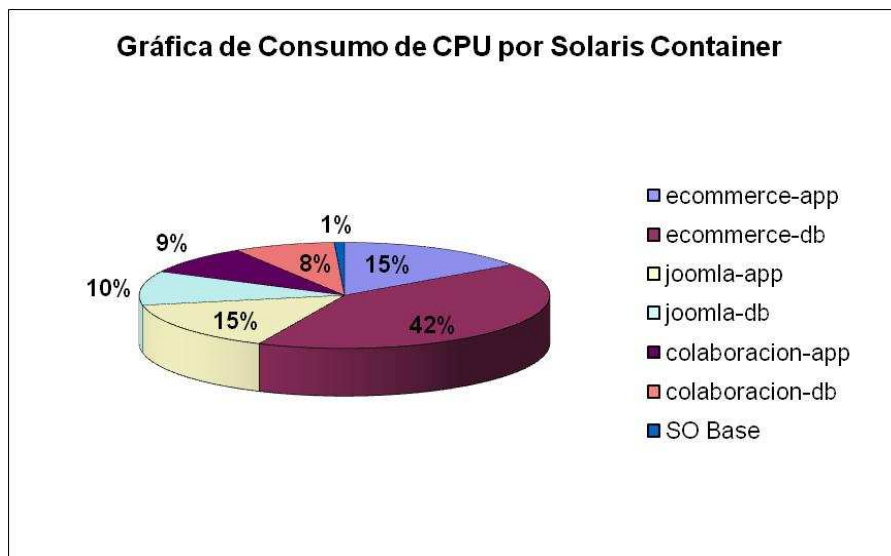


Grafica 5.4 Uso de recursos

Al iniciar la prueba el consumo de CPU y Memoria RAM se eleva considerablemente a comparación de los resultados obtenidos anteriormente.

El promedio obtenido del porcentaje de CPU utilizado es 20.25% y de Memoria RAM es 53.91%.

Tomando el porcentaje de uso de CPU, observamos que este se divide en los Solaris Containers, por lo que con ayuda del monitoreo implementado, se obtuvo que este porcentaje se divide de la forma que se muestra en la gráfica 5.5.



Grafica 5.5 Consumo de CPU

La aplicación de Comercio Electrónico es la que más consume recursos de CPU y Memoria RAM, seguida por el Manejador de Contenidos y el Sitio de Colaboración. Mientras que el Sistema Operativo Base solo consume 1% de los recursos del servidor.

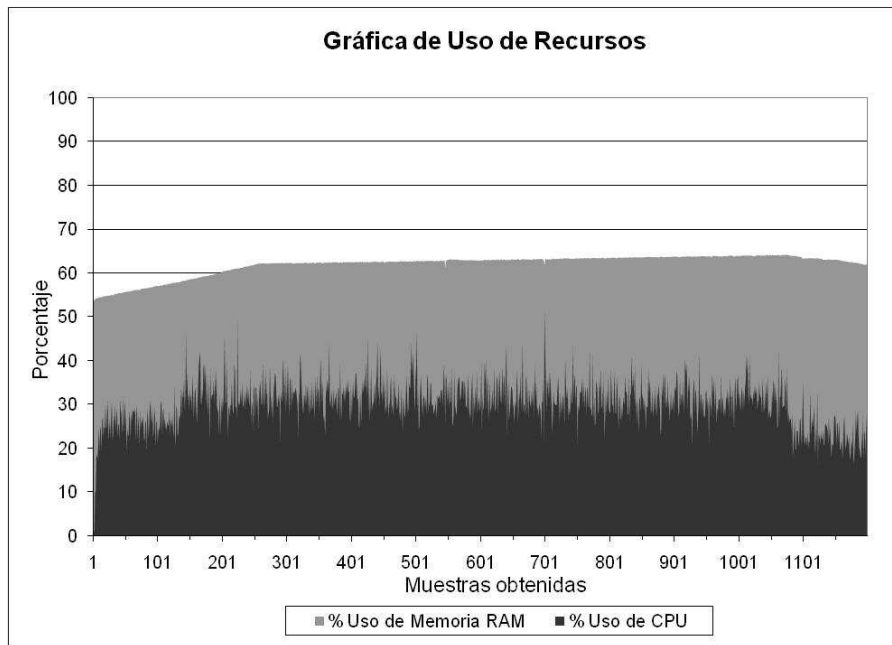
**Prueba 3: Generar el doble de carga hacia cada aplicación con la herramienta Apache JMeter.**

**Objetivo:** Evaluar el uso de recursos de CPU y Memoria RAM en el Sistema Operativo Base.

**Realización de la prueba**

Se configuro la herramienta Apache JMeter para generar 50000 peticiones para las aplicaciones de Comercio Electrónico y Manejador de contenidos con una duración de 5 horas, esto sería equivalente a 10000 usuarios por hora. Para el Sitio de Colaboración se configuraron 2000 usuarios con duración de 5 horas, equivalente a 400 usuarios por hora.

**Resultados**



Grafica 5.6 Uso de recursos

En la Grafica 5.6 se observa que el promedio obtenido del porcentaje de CPU utilizado es 30.22% y de Memoria RAM es 63%.

La tabla 5.2 contiene un resumen de los resultados obtenidos para la evaluación del aprovechamiento de los recursos.

	Sin Virtualización		Con Virtualización		
	Servidor y Sistema Operativo	Servidor con aplicación Comercio Electrónico	Servidor con Solaris Containers	Servidor con Solaris Containers y Aplicaciones	Servidor con Solaris Containers y Aplicaciones con carga de trabajo doble
<b>% Uso de CPU</b>	0.26	6.64	0.53	20.25	30.22
<b>% Uso de Memoria RAM</b>	29.77	31.84	37.14	53.91	63

Se puede observar el aprovechamiento máximo de los recursos del servidor cuando se utiliza una arquitectura con Virtualización. Ambos casos presentados en la tabla (Sin virtualización y Con Virtualización), son sobre el mismo servidor Sun Fire T2000 y en ambos casos, se cuenta con la misma configuración de hardware.

Sin Virtualización los recursos del servidor no se aprovechan, se tiene un procesador de 4 cores trabajando a un máximo del 6.64% y 8GB de RAM a un máximo del 31.84%. Estos datos se recopilaron del servidor sin virtualizar. Viendo la siguiente columna, podemos apreciar a simple vista que con los ambientes virtualizados mediante Solaris Containers se aprovechan más los recursos de hardware. Se observa que el máximo uso del procesador es al 30.22%, más del doble que con una arquitectura sin virtualizar. Y el uso máximo de la memoria RAM es un 63%, un poco más del doble que sin virtualización.

Analizando la tabla, podemos concluir que nuestra solución de Virtualización mediante Solaris Containers, genera un mejor aprovechamiento de los recursos de hardware del servidor y ningún componente se subutiliza, aprovechando todos los recursos con los que se cuenta.

### 5.3.2 Costo total de propiedad o Total Costo of Ownership (TCO)

Como ya se ha mencionado a lo largo de esta Tesis, el Coste Total de Propiedad (TCO en sus siglas en inglés) es un tema muy importante de negocios, ya que una vez entendido se vuelve un factor crítico para identificar el Retorno de Inversión (ROI por sus siglas en inglés) de un proyecto de TI.

Cuando una organización o una compañía está desarrollando un proyecto, siempre se toma en cuenta el valor del TCO cuando se empieza a tocar el delicado tema del gasto que representará para la compañía, el llevar a cabo un proyecto. Para poder medir el éxito de un proyecto se debe de medir el TCO antes de llevar a cabo el proyecto y después de un tiempo ya implementado, para poder apreciar el coste real que representó el proyecto.

Medir el TCO no es fácil y la mayoría de las veces depende de los valores que realmente le interesen a la compañía, es decir, una organización o compañía debe de preguntarse cuáles valores son importantes para ella, como pueden ser el ahorro de energía, el costo de espacio, el costo de administración de los equipos, etc.

Existen varios modelos de TCO, entre los cuales se encuentran:

- Coste Total de Propiedad (TCO). Un concepto desarrollado por Gartner, es una metodología que se enfoca en los costes del entorno de TI (espacio, energía, recursos, etc).
- Impacto Económico Total (TEI). Desarrollado por Giga Information Group, se enfoca en cómo una solución impacta todas las áreas de negocio de una organización.
- Valor Total de Propiedad (TVO). Desarrollado por Lucidus Ltd. Esta metodología tiene como principal objetivo, no sólo medir el impacto de un proyecto, sino también medir el valor agregado que aporta el proyecto a la organización.

Para este análisis, nos basamos en el modelo de Gartner para medir el TCO que representa el proyecto.

Gartner define el Coste Total de Propiedad (en adelante TCO), como todo el coste asociado a un proyecto de TI, es decir, no sólo toma en cuenta el coste del *hardware*, también toma en cuenta otros costes que van ligados a éste, como son: el costo de adquisición del *hardware*, la instalación, la implementación, la administración, el mantenimiento e incluso la desinstalación del *hardware*. Normalmente, el costo de

adquisición del hardware representa un 15% del TCO y el resto corresponde al costo de administración e implementación del proyecto en su totalidad.

El modelo de Gartner define 2 categorías para definir costo: Costos Directos y Costos Indirectos. Dentro de la primera categoría, se encuentran el costo del *hardware* y del *software*, administración de sistemas, almacenamiento, soporte, recursos, servicios de consultoría, etc. Los Costos Indirectos abarcan la parte del servicio que ofrece la solución del proyecto de TI, como es la disponibilidad y el *downtime* en caso de falla del sistema.

Para realizar el análisis de TCO, existe una herramienta desarrollada por Gartner llamada *Gartner's TCO Manager*, la cual contiene una extensa base de datos recolectada de 300 centros de datos de 200 compañías diferentes. La herramienta tiene un costo de licencia, no existen versiones de demostración; pero la recomendamos ampliamente para aquellas organizaciones que cuenten con los recursos necesarios.

Realizamos un análisis de TCO basándonos en los siguientes supuestos:

- El siguiente análisis de TCO es un análisis simple de los ahorros que se generan al consolidar servidores mediante la virtualización.
- El inventario usado como base del análisis está compuesto por 3 servidores físicos, que son los que se tomarán en cuenta para la consolidación mediante virtualización.
- Los servidores del inventario incluye las marcas Sun Microsystems.
- El servidor Sun Fire T2000 es la plataforma usada para la virtualización. El costo del servidor y su configuración es de \$60,000 USD. Este costo es un estimado y no incluye descuentos.
- Todos los datos del análisis están basados en precios estimados. Los costos reales pueden diferir con los de este análisis.
- Los resultados obtenidos de este análisis representan un aproximado del ahorro que se puede lograr mediante la consolidación de servidores, pero no garantiza que los datos aquí obtenidos se cumplan.

Como primer paso, se obtiene el análisis de TCO de acuerdo a la arquitectura actual, es decir, antes de la virtualización de servidores. De esta manera después podremos hacer una comparativa de costos entre la situación actual y la propuesta.

Todos los datos contenidos en las tablas están expresados en dólares americanos y en tiempo corresponden a 1 año.

- Tabla 5.3 Análisis de TCO del Escenario No Virtualizado

Costos Directos	Año 1	Total (\$)	% del Costo Total
<b>Hardware</b>			
Servidores	\$ 403,200.00	\$ 403,200.00	45.4%
Consolas	\$ 10,000.00	\$ 10,000.00	1.1%
Periféricos	\$ -	\$ -	0.0%
Componentes de Red	\$ 10,000.00	\$ 10,000.00	1.1%
Costo de Mantenimiento	\$ 48,384.00	\$ 48,384.00	5.4%
Energía (Enfriamiento y Alimentación)	\$ 5,936.00	\$ 5,936.00	0.7%
<b>Costo total de Hardware</b>	<b>\$ 477,520.00</b>	<b>\$ 477,520.00</b>	<b>53.7%</b>
<b>Software</b>			
Licencias	\$ -	\$ -	0.0%
Costo de Mantenimiento	\$ -	\$ -	0.0%
<b>Costo total de Software</b>	<b>\$ -</b>	<b>\$ -</b>	<b>0.0%</b>
<b>Administración</b>			
Red	\$ 20,000.00	\$ 20,000.00	2.3%
Servidores	\$ 60,000.00	\$ 60,000.00	6.8%
Almacenamiento	\$ 20,000.00	\$ 20,000.00	2.3%
Bases de Datos	\$ 20,000.00	\$ 20,000.00	2.3%
Servidores de Aplicaciones	\$ 20,000.00	\$ 20,000.00	2.3%
<b>Costo total de Administración</b>	<b>\$ 140,000.00</b>	<b>\$ 140,000.00</b>	<b>15.8%</b>
<b>Soporte</b>			
Personal de Soporte	\$ 60,000.00	\$ 60,000.00	6.8%
Entrenamiento	\$ 9,000.00	\$ 9,000.00	1.0%
Adquisiciones	\$ -	\$ -	0.0%
Viáticos	\$ -	\$ -	0.0%
Contratos de Soporte	\$ 90,000.00	\$ 90,000.00	10.1%
Actividades Extras	\$ -	\$ -	0.0%
<b>Costo total de Soporte</b>	<b>\$ 159,000.00</b>	<b>\$ 159,000.00</b>	<b>17.9%</b>
<b>Implementación</b>			
Desarrollo/Personalización/Integración	\$ 30,000.00	\$ 30,000.00	3.4%
Entrenamiento	\$ -	\$ -	0.0%
Consultoría	\$ -	\$ -	0.0%
Otros	\$ -	\$ -	0.0%
<b>Costo total de Implementación</b>	<b>\$ 30,000.00</b>	<b>\$ 30,000.00</b>	<b>3.4%</b>
<b>Telecomunicaciones</b>			
Red Local	\$ 1,350.00	\$ 1,350.00	0.2%
Red Externa	\$ 36,000.00	\$ 36,000.00	4.1%
Acceso Remoto	\$ -	\$ -	0.0%
<b>Costo total de Telecomunicaciones</b>	<b>\$ 37,350.00</b>	<b>\$ 37,350.00</b>	<b>4.2%</b>
<b>Total de Costos Directos</b>	<b>\$ 843,870.00</b>	<b>\$ 843,870.00</b>	<b>94.9%</b>



Costos Indirectos	Año 1	Total (\$)	% del Costo Total
<b>Usuario Final</b>			
Entrenamiento del Usuario Final	\$ -	\$ -	0.0%
Entrenamiento informal del usuario	\$ -	\$ -	0.0%
Desarrollo en Casa de aplicaciones	\$ -	\$ -	0.0%
Mantenimiento de Archivos	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo total de Usuario Final</b>	<b>\$ -</b>	<b>\$ -</b>	<b>0.0%</b>
<b>Tiempo fuera de línea</b>			
Tiempo fuera de línea planeado	\$ 15,000.00	\$ 15,000.00	1.7%
Tiempo fuera de línea no planeado	\$ 30,000.00	\$ 30,000.00	3.4%
...	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo total del tiempo fuera de línea</b>	<b>\$ 45,000.00</b>	<b>\$ 45,000.00</b>	<b>5.1%</b>
<b>Total de Costos Indirectos</b>	<b>\$ 45,000.00</b>	<b>\$ 45,000.00</b>	<b>5.1%</b>

Costos Directos	Año 1	Total (\$)	% del Costo Total
<b>Costo total de Hardware</b>	<b>\$ 477,520.00</b>	<b>\$ 477,520.00</b>	<b>53.7%</b>
<b>Costo total de Software</b>	<b>\$ -</b>	<b>\$ -</b>	<b>0.0%</b>
<b>Costo total de Administración</b>	<b>\$ 140,000.00</b>	<b>\$ 140,000.00</b>	<b>15.8%</b>
<b>Costo total de Soporte</b>	<b>\$ 159,000.00</b>	<b>\$ 159,000.00</b>	<b>17.9%</b>
<b>Costo total de Implementación</b>	<b>\$ 30,000.00</b>	<b>\$ 30,000.00</b>	<b>3.4%</b>
<b>Costo total de Telecomunicaciones</b>	<b>\$ 37,350.00</b>	<b>\$ 37,350.00</b>	<b>4.2%</b>
<b>Total de Costos Directos</b>	<b>\$ 843,870.00</b>	<b>\$ 843,870.00</b>	<b>94.9%</b>
<b>Costos Indirectos</b>			
<b>Costo total de Usuario Final</b>	<b>\$ -</b>	<b>\$ -</b>	<b>0.0%</b>
<b>Costo total del tiempo fuera de línea</b>	<b>\$ 45,000.00</b>	<b>\$ 45,000.00</b>	<b>5.1%</b>
<b>Total de Costos Indirectos</b>	<b>\$ 45,000.00</b>	<b>\$ 45,000.00</b>	<b>5.1%</b>
<b>Total Costos</b>	<b>\$ 888,870.00</b>	<b>\$ 888,870.00</b>	

La tabla 5.3 refleja el costo total que representa una arquitectura no virtualizada para nuestro caso de estudio.

Se toman en cuenta datos que implican un costo y que tienen que ver con el hardware. Por ejemplo: los recursos de administración del centro de datos, los recursos de soporte para los servidores, el entrenamiento del personal, etc.

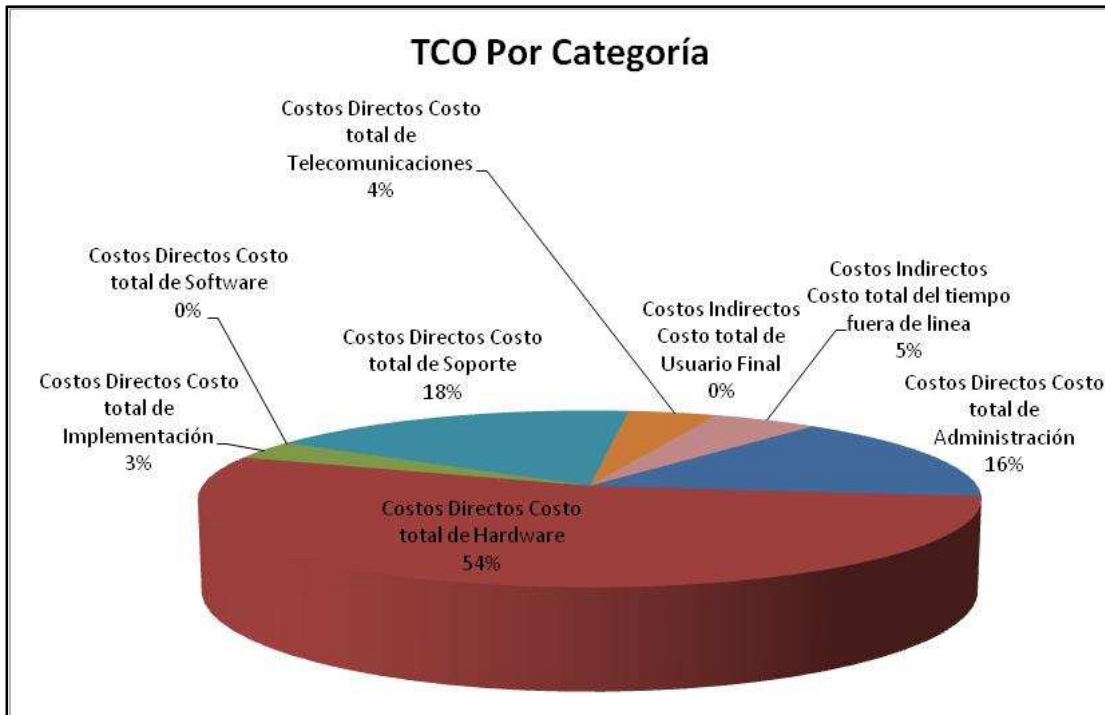
Como se mencionó al inicio de este subtema, existen dos tipos de costo según el modelo de Gartner: los directos y los indirectos.

Alrededor del hardware existen diferentes tipos de costo directo que deben de tomarse en cuenta, ya que pueden ser representativos al momento de generar un análisis de Costo Total.

Se recomienda tomar en cuenta costos como los que se abarcan en la tabla: costo de adquisición del hardware, otro equipo relacionado con él (consolas, computadoras de escritorio, switches, cables, etc), licencias de software si es que aplica. En nuestro caso, no se incluyen costos de licenciamiento de software ya que todo el software usado en la Tesis es libre y gratuito; pero si se llega a trabajar con un licenciamiento de software, deberá de incluirse en el TCO.

En la tabla observamos que el costo anual de una arquitectura que comprende 3 servidores sin virtualizar, tiene un costo total anual de \$888,870.00 USD. Es decir, tomando en cuenta que cada servidor contiene únicamente una aplicación, más todos los costos que estos generan a su vez (mantenimiento, administración, soporte, etc) representan un gasto anual de más de medio millón de dólares.

Enseguida se muestra la gráfica 5.13 en donde se aprecia la proporción del costo total de propiedad por cada categoría considerada, considera tanto costos directos como costos indirectos. Es notorio que en el caso de los servidores no virtualizados, en un horizonte de un año, el costo del hardware representa más del 50% del costo total de propiedad, aspecto que ha de considerarse para el análisis del costo total de propiedad con servidores virtualizados.



Grafica 5.13 Escenario No Virtualizado

En el siguiente apartado, se lleva a cabo el análisis de Costo Total pero para la arquitectura propuesta en esta Tesis, con virtualización mediante Solaris Containers. Se puede apreciar claramente una diferencia entre ambos análisis presentados.

- **Tabla 5.4 Análisis de TCO con Virtualización**

Costos Directos	Año 1	Total (\$)	% de Costo Total
<b>Hardware</b>			
Servidores	\$ 134,400.00	\$ 134,400.00	26.8%
Clientes	\$ 10,000.00	\$ 10,000.00	2.0%
Periféricos	\$ -	\$ -	0.0%
Componentes de Red	\$ 10,000.00	\$ 10,000.00	2.0%
Mantenimiento	\$ 16,128.00	\$ 16,128.00	3.2%
Costo de energía	\$ 1,978.85	\$ 1,978.85	0.4%
<b>Costo Total de Hardware</b>	<b>\$ 172,506.85</b>	<b>\$ 172,506.85</b>	<b>34.4%</b>
<b>Software</b>			
Licencia	\$ -	\$ -	0.0%
Mantenimiento		\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo Total de Software</b>	<b>\$ -</b>	<b>\$ -</b>	<b>0.0%</b>
<b>Administración</b>			
Redes	\$ 20,000.00	\$ 20,000.00	4.0%
Sistemas	\$ 20,000.00	\$ 20,000.00	4.0%
Almacenamiento	\$ 20,000.00	\$ 20,000.00	4.0%
DBA	\$ 20,000.00	\$ 20,000.00	4.0%
Aplicativos	\$ 20,000.00	\$ 20,000.00	4.0%
<b>Costo Total de Administración</b>	<b>\$ 100,000.00</b>	<b>\$ 100,000.00</b>	<b>19.9%</b>
<b>Soporte</b>			
Recursos de Soporte	\$ 30,000.00	\$ 30,000.00	6.0%
Entrenamiento	\$ 3,000.00	\$ 3,000.00	0.6%
Procurement procedures	\$ -	\$ -	0.0%
Viajes	\$ -	\$ -	0.0%
Contratos de Soporte	\$ 20,000.00	\$ 20,000.00	4.0%
Horas Extra	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo Total de Soporte</b>	<b>\$ 53,000.00</b>	<b>\$ 53,000.00</b>	<b>10.6%</b>
<b>Implementación</b>			
Desarrollo/customización/integración	\$ 20,000.00	\$ 20,000.00	4.0%
Entrenamiento		\$ -	0.0%
Consultoría/ Servicios de Integración	\$ 30,000.00	\$ 30,000.00	6.0%
Otro	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo Total de Desarrollo</b>	<b>\$ 50,000.00</b>	<b>\$ 50,000.00</b>	<b>10.0%</b>
<b>Gastos de Comunicación</b>			
Local Area Network	\$ 450.00	\$ 450.00	0.1%
Wide Area Network	\$ 36,000.00	\$ 36,000.00	7.2%
Acceso Remoto	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Gastos Totales de Comunicación</b>	<b>\$ 36,450.00</b>	<b>\$ 36,450.00</b>	<b>7.3%</b>
<b>Costos Totales Directos</b>	<b>\$ 411,956.85</b>	<b>\$ 411,956.85</b>	<b>82.1%</b>

Costos Indirectos	Año 1	Total (\$)	% de Costo Total
<b>Usuarios Finales</b>			
Entrenamiento Formal		\$ -	0.0%
Entrenamiento Informal		\$ -	0.0%
Auto desarrollo de aplicaciones	\$ -	\$ -	0.0%
Mantenimiento de Archivos Locales	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo Total de Usuarios Finales</b>	<b>\$ -</b>	<b>\$ -</b>	<b>0.0%</b>
<b>Baja de Servicio</b>			
Baja de Servicio Programada	\$ 30,000.00	\$ 30,000.00	6.0%
Baja de Servicio no Programada	\$ 60,000.00	\$ 60,000.00	12.0%
...	\$ -	\$ -	0.0%
...	\$ -	\$ -	0.0%
<b>Costo Total de Baja de Servicio</b>	<b>\$ 90,000.00</b>	<b>\$ 90,000.00</b>	<b>17.9%</b>
<b>Costos Totales Indirectos</b>	<b>\$ 90,000.00</b>	<b>\$ 90,000.00</b>	<b>17.9%</b>
<b>Costo Total</b>	<b>\$ 501,956.85</b>	<b>\$ 501,956.85</b>	

En la tabla 5.4 se realizó un análisis y se puede apreciar que el costo total al año de una solución virtualizada, es de \$501,956.85 USD. Se tomaron en cuenta valores de costo de hardware, entre los que se encuentran: el costo de adquisición del servidor usado para virtualizar, que en nuestro caso se trata de un servidor Sin Fire T2000. También se toman otros valores de hardware como las computadoras de escritorio, equipo de redes de datos, costo de energía del hardware y el mantenimiento del mismo.

Para calcular el costo de energía nos basamos en el consumo de “placa” del Servidor Sun Fire T2000, el cual está especificado en una hoja técnica del servidor y que también se encuentra en el Anexo 1 de este trabajo de Tesis. Ya contando con el valor del consumo de energía del servidor en Watts, este valor se multiplica por las horas de funcionamiento del mismo, es decir:

$$(W \times 24 [h]) = kW/h$$

De esta manera obtenemos el valor de consumo en kilowatts del servidor por hora. Después basándonos en el costo de la energía eléctrica en nuestro país, el cual obtuvimos de la página oficial de Comisión Federal de Electricidad, el costo del kilowatt por hora es de un promedio de 3.285 pesos.

Dado que nuestro análisis de TCO está expresado en dólar americano (USD), el costo del kilowatt por hora es de 0.2657 USD. Por lo que el consumo de energía del servidor

Sun Fire T2000 al año es de \$1047.692 USD. A esto se debe de sumar la energía consumida por el aire acondicionado, el cual se expresa normalmente en unidades de medida llamadas BTU.

Continuando con el análisis de TCO con Virtualización mostrado en la tabla anterior, también se toman en cuenta los costos de los recursos de administración y soporte de un centro de datos, como son los Administradores de Sistemas, los Administradores de Bases de Datos (DBA), los Administradores de Redes, etc. Cada recurso o persona asignada a una tarea, también representa un costo a la organización.

En la tabla se puede apreciar que los costos de Software están en ceros. Esto se debe a que nuestro caso de uso contempla software libre y gratuito, lo que no genera un impacto en los costos.

Otros datos que se tomaron en cuenta para este análisis de TCO, son el costo del soporte del hardware, la implementación de la solución virtualizada (Instalación de Sistema Operativo, Configuración, creación de containers, etc), servicios de consultoría, entrenamiento de los recursos internos, etc.

En la sección de Costos Indirectos existen dos valores que se identifican como “Baja de Servicio Programada” y “Baja de Servicio No Programada”. Esto se incluye ya que siempre que se realiza una migración, una renovación tecnológica o un cambio en la arquitectura actual del centro de datos o de un servicio en particular, se corre un riesgo, el cual siempre tendrá un impacto en la producción del centro de datos. Aún sin llevar a cabo un cambio en el centro de datos o en la arquitectura de un servicio en particular, siempre existe el riesgo de una falla.

Por esto mismo, deben de tomarse en cuenta estos valores, ya que toda baja de servicio representa un costo a la organización y cuando se lleva a cabo una modificación en la arquitectura, el impacto puede ser mayor a lo normal.

Comparando la tabla de “Análisis de Costo Actual” con la tabla “Análisis de Costo con Virtualización”, se observa que dentro de los costos indirectos existe una diferencia en los valores de cada tabla y los costos de “Baja de Servicio” son mayores en la segunda tabla; esto es por la virtualización de los servicios.

Esto no significa que cada año se vayan a tener estos costos como impacto de la virtualización. Los datos de la tabla únicamente representan el primer año que es cuando pueden surgir problemas de mala configuración que requieran dar de baja por un tiempo el servicio, falta de entrenamiento de los recursos de soporte, falta de actualizaciones, etc.

Todos estos son problemas comunes pero que pueden corregirse, y esto se debe de tomar en cuenta en un análisis de TCO.

A continuación se muestra el resumen de las gráficas del Análisis de TCO con Virtualización.

**Gráficas del Resumen del Análisis de TCO con Virtualización**

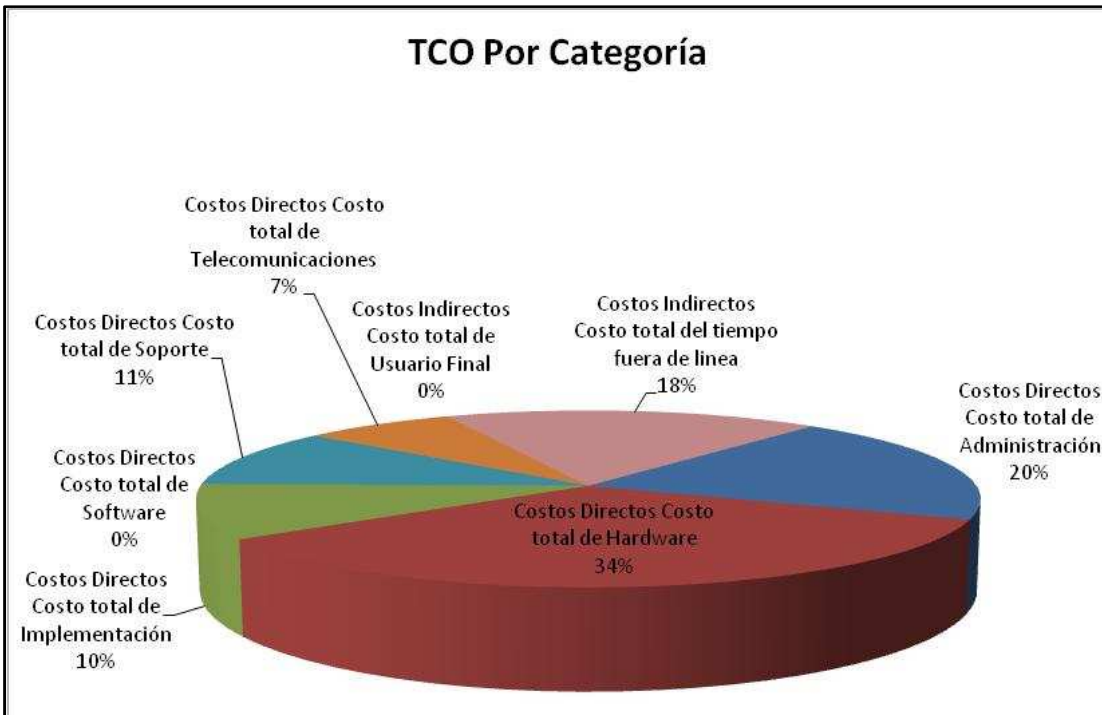
Tabla 5.5

<b>Costos Directos</b>	<b>% del Costo Total</b>
Total Hardware Costs	34.4%
Total Software Costs	0.0%
Total Management Costs	19.9%
Total Support Costs	10.6%
Total Development Costs	10.0%
Total Communication Fees	7.3%
<b>Costo Directo Total</b>	<b>82.1%</b>

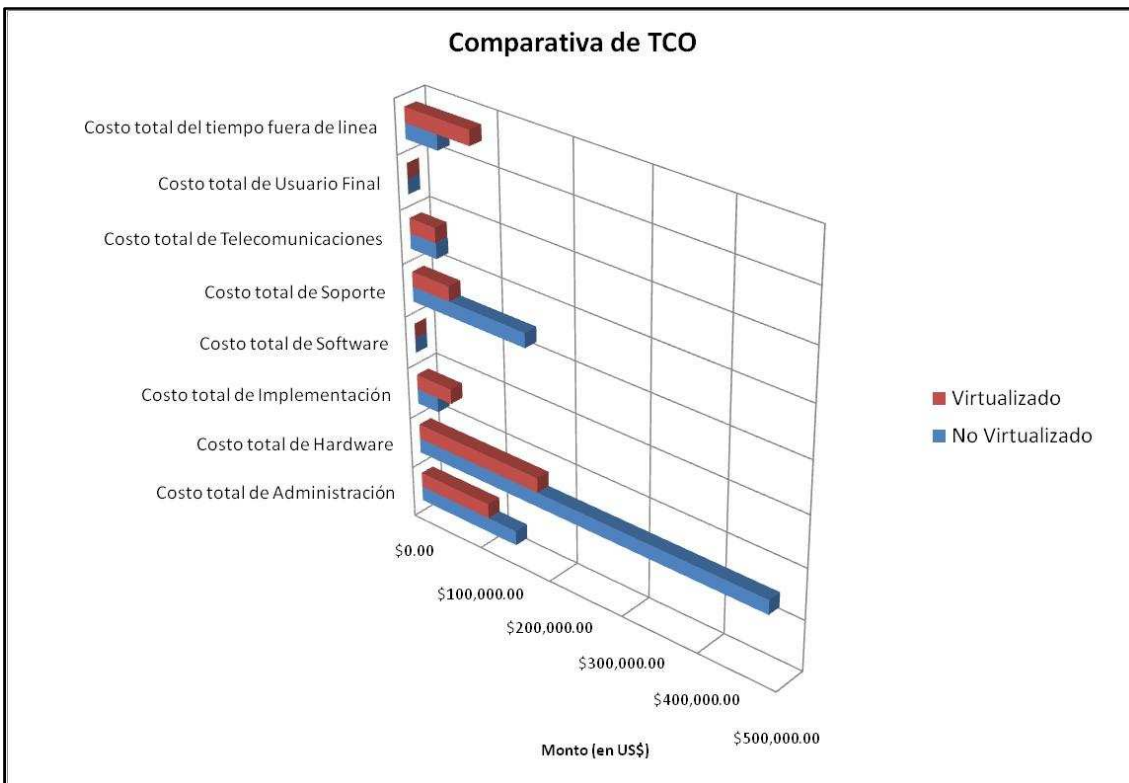
<b>Costo Indirecto</b>	<b>% del Costo Total</b>
Total End User IS costs	0.0%
Total Downtime costs	17.9%
<b>Costo Total Indirecto</b>	<b>17.9%</b>

	<b>Año 1</b>
Total Direct Costs	\$ 411,956.85
Total Indirect Costs	\$ 90,000.00

Tal como se mostró en el escenario de servidores No Virtualizados, en la gráfica 5.14 se muestra la comparativa por categoría en el costo total de propiedad. En proporción, el costo de adquisición de hardware bajó de más del 50% en el escenario No Virtualizado a 34% en el escenario Virtualizado, sin embargo el ahorro real se aprecia en la gráfica 5.15, donde se compara categoría por categoría entre ambos escenarios.



Grafica 5.14 Escenario virtualizado



Grafica 5.15 Comparativa de TCO



En base a la gráfica 5.15 el Análisis de TCO No Virtualizado contra el Análisis de TCO con Virtualización, puede apreciarse a simple vista que el costo de la solución con virtualización tiene un ahorro de aproximadamente \$400,000 USD.

El resultado del análisis mostrado comprueba que una arquitectura virtualizada genera un ahorro sustancial del costo total de un centro de datos.

### 5.3.3 Espacio físico

Se obtienen las especificaciones del servidor del documento Guía de planificación de la instalación Servidor Sun SPARC Enterprise T2000, donde se proporcionan las especificaciones físicas, dimensiones de altura, ancho, profundidad y peso, así como los requerimientos para la correcta instalación del servidor.

Especificaciones físicas		
Dimensiones	EE.UU.	Sistema métrico
Ancho	17,3 pulgadas	440 mm
Profundidad	24,3 pulgadas	617 mm
Altura (2 unidades de bastidor)	3,5 pulgadas	89 mm
Peso aproximado (sin tarjetas PCI ni piezas de montaje en el bastidor)	40 lb	18 kg

### Resultados



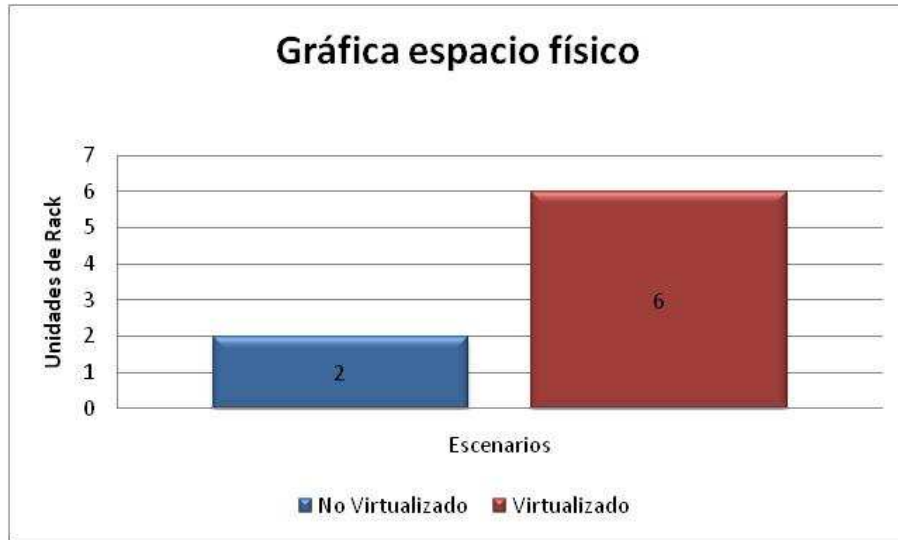
Fig. 5.7 Consumo de energía

Figura 5.8 Consumo de energía

**Análisis de Resultados**

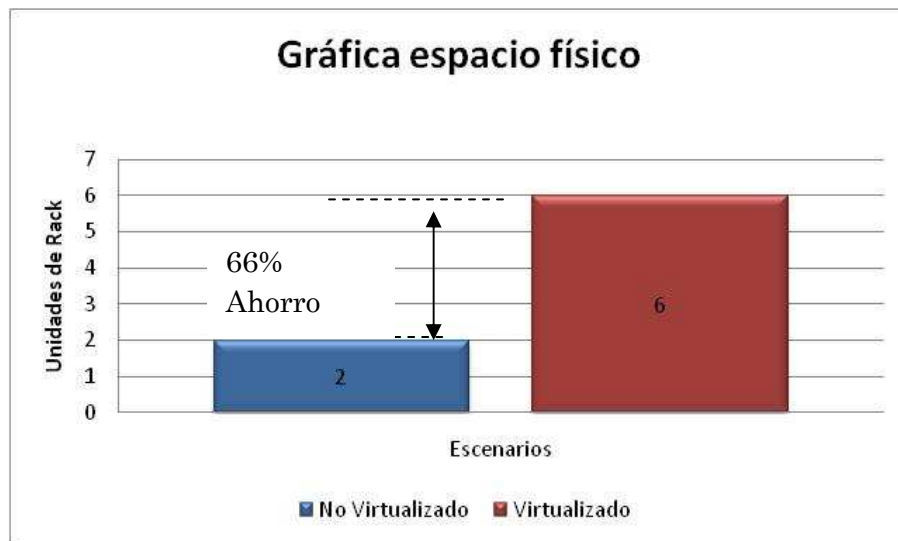
La figura 5.7 muestra que cada servidor ocupa 2U (Unidades de Rack) o 89 mm, entonces con los 3 servidores se ocupa 6 U (Unidades de Rack).

La figura 5.8 muestra que en este escenario gracias a la virtualización, se consolidan los servidores en un solo equipo, cuyo espacio en Rack es de solamente 2U (Unidades de Rack).



Grafica 5.7 Espacio físico en unidades de rack

Se observa en la gráfica 5.7 que un escenario no Virtualizado, los servidores en el Rack ocupan más del doble de espacio que un escenario Virtualizado.



Grafica 5.8 Porcentaje de espacio ahorrado

En la gráfica 5.8 se muestra que en los escenarios Virtualizados representan un ahorro del **66%** de espacio que se ocupa en un Rack.

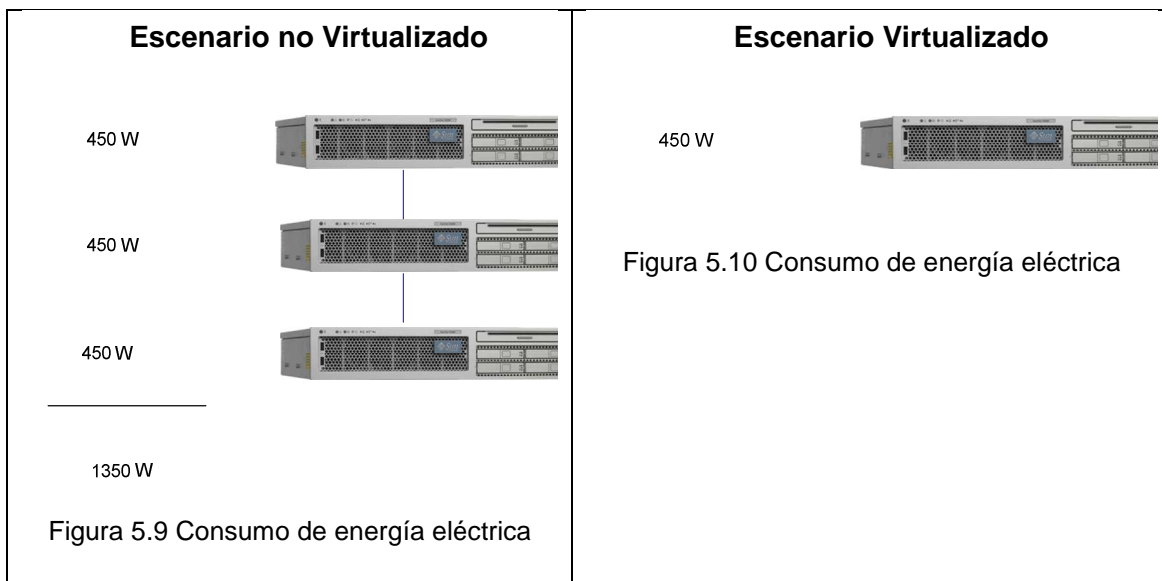
### 5.3.4 Energía eléctrica consumida

Se obtienen las especificaciones del servidor del documento Guía de planificación de la instalación Servidor Sun SPARC Enterprise T2000, donde se proporcionan las especificaciones de voltaje y potencia, así como los requerimientos para la correcta instalación del servidor.

Sun SPARC Enterprise T2000

Descripción	Especificación
Escala de voltajes de entrada operativos	De 100 a 240 VCA, 50-60 Hz
Máxima intensidad de corriente de entrada operativa	4,5 A a 100 - 120 VCA 2,25 A a 200 - 240 VCA
Máxima potencia de entrada operativa	450 W
Máxima disipación térmica	1.365 BTU/h (1.440 KJ/h)

## Resultados



### Escenario no Virtualizado

En la figura 5.9 se observa que en este escenario se tienen 3 servidores, cada uno consume 450W y por lo tanto el consumo total de estos servidores es de **1350 W/h**.

Se obtiene el costo promedio del año 2010 para la ciudad de México, el cual es de \$3.2805 pesos por KW/h y correspondiente a U\$ 0.2657 KWh y se tiene que cada servidor consume 450 Wh entonces se obtiene:

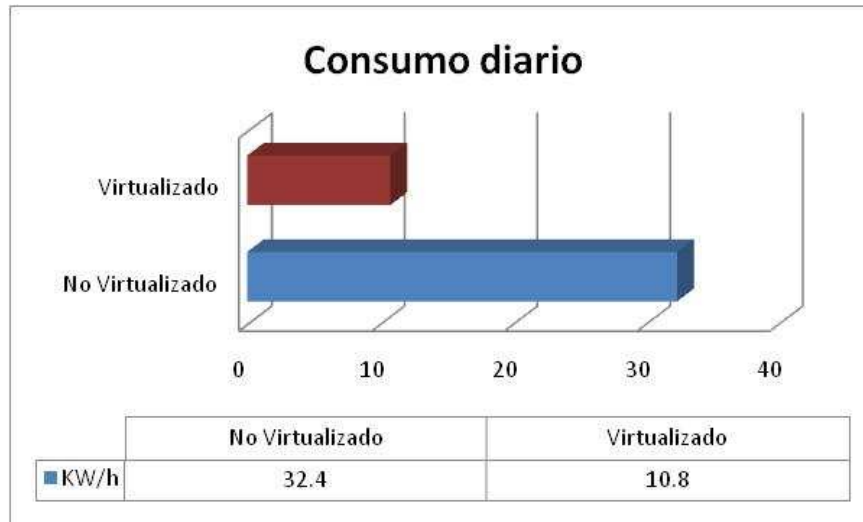
- Consumo diario de **32.4 KW/h**
- Consumo Anual de **11826 KW/h**
- Costo Anual de **U\$3,142**

### Escenario Virtualizado

En la figura 5.10 se observa que en este escenario gracias a la virtualización, se consolidan los servidores en un solo equipo, cuyo consumo total es de 450Wh

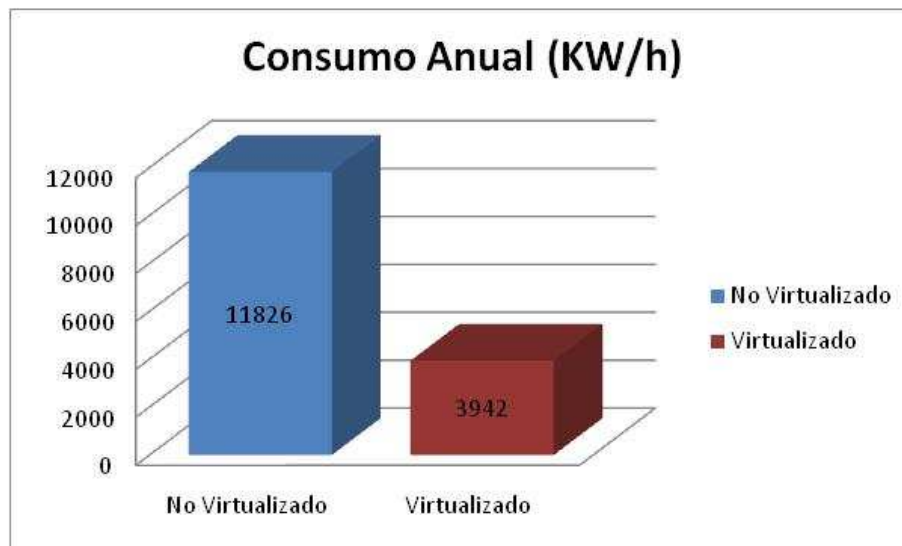
- Consumo diario 32.4 KW/h
- Consumo Anual de 3942 KW/h
- Costo Anual de U\$1,048

### Análisis de Resultados



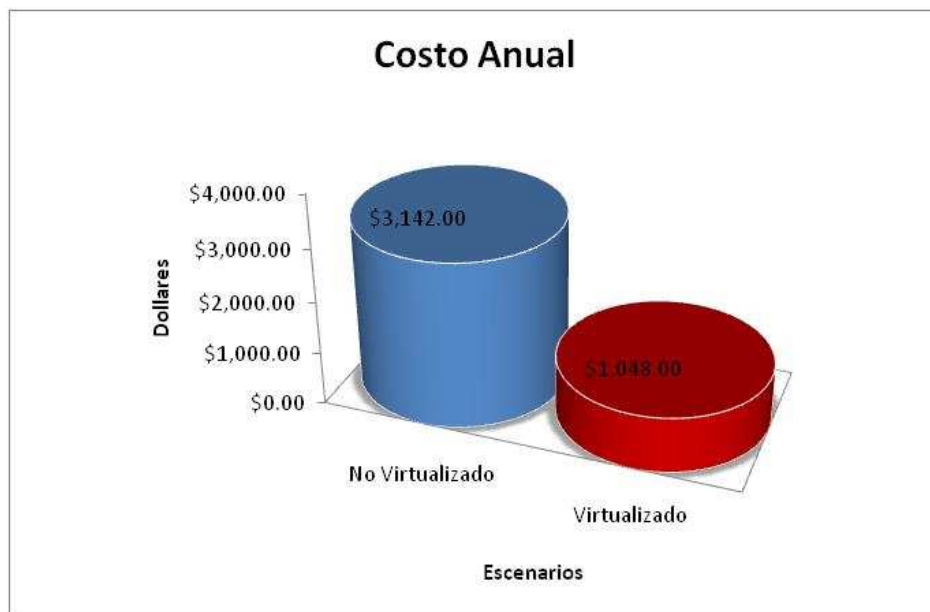
Grafica 5.9 Consumo diario de los servidores

En la gráfica 5.9 se observa que el consumo diario es mayor del 50% en escenario sin virtualización, que uno virtualizado.



Grafica 5.10 Consumo anual de los servidores

Se muestra en la gráfica 5.10 que en el escenario sin virtualizar el consumo es muy alto en comparación con uno virtualizado. En el ambiente virtualizado se ahorran 7884 KW/h Anuales y esto se ve directamente reflejado en los costos asociados con la energía.



Grafica 5.11 Consumo anual

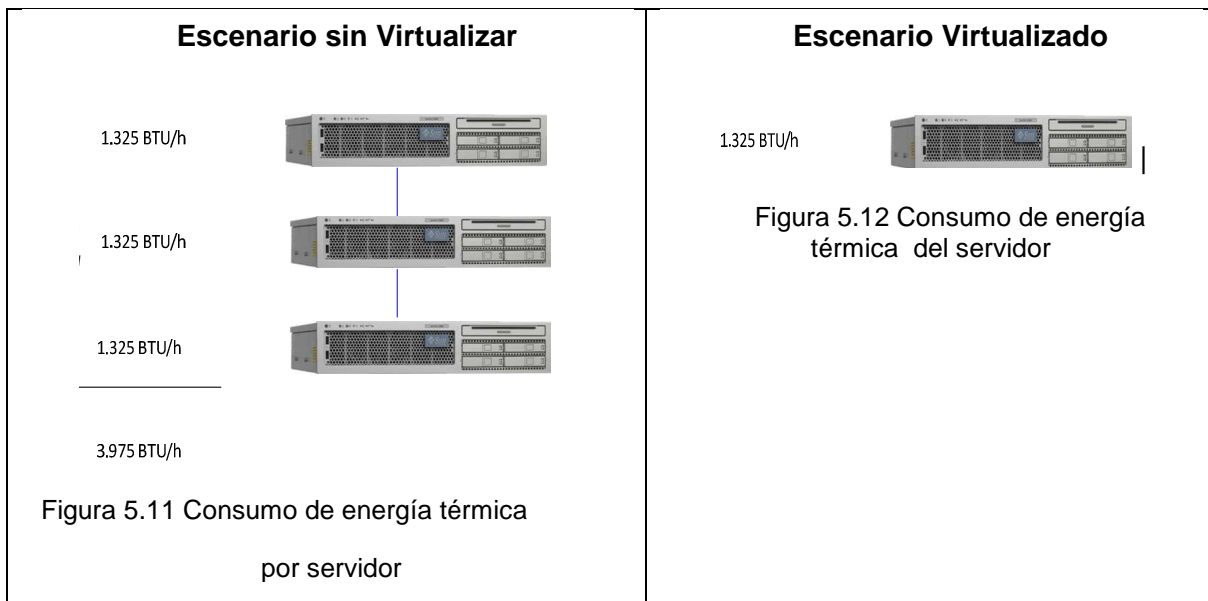
De la gráfica 5.11 se puede inferir que el ahorro de energía es de U\$2,094 anuales, lo que representa un ahorro de más del 50% con la implementación de un escenario virtualizado.

### 5.3.5 Energía Térmica

Se obtienen las especificaciones del servidor del documento Guía de planificación de la instalación Servidor Sun SPARC Enterprise T2000, donde se proporcionan las especificaciones de disipación térmica.

Descripción	Especificación
Escala de voltajes de entrada operativos	De 100 a 240 VCA, 50-60 Hz
Máxima intensidad de corriente de entrada operativa	4,5 A a 100 - 120 VCA 2,25 A a 200 - 240 VCA
Máxima potencia de entrada operativa	450 W
Máxima disipación térmica	1.365 BTU/h (1.440 KJ/h)

### Resultados



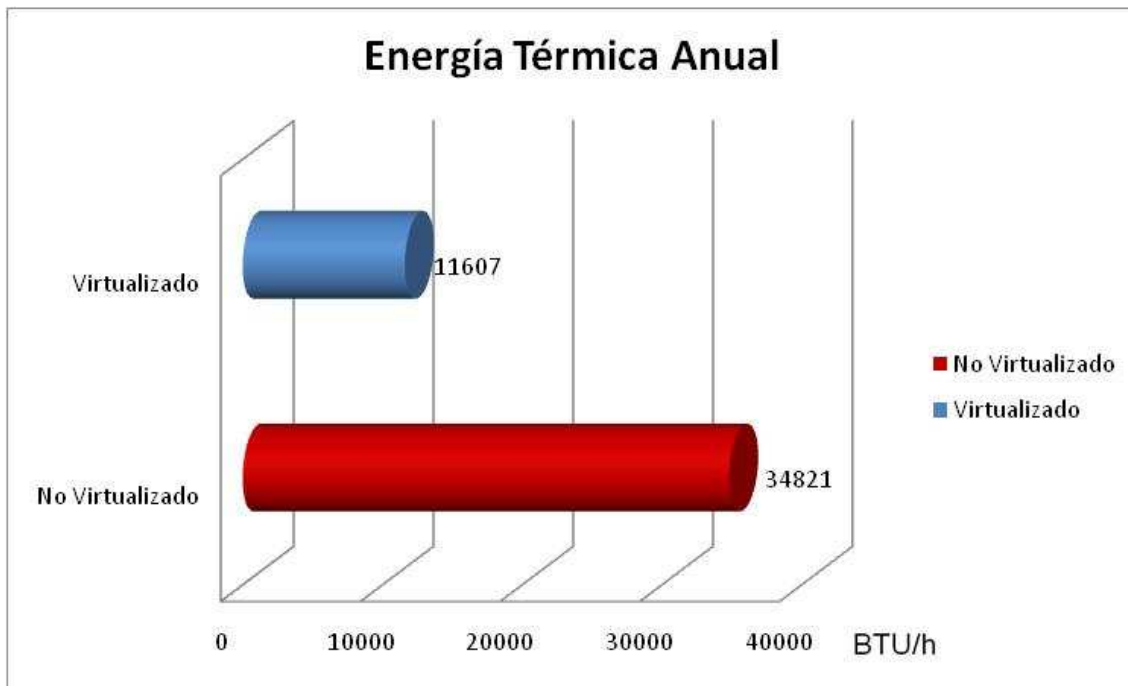
**Escenario Virtualizado**

En este escenario gracias a la virtualización, se consolidan los 3 servidores en un solo equipo (Figura 5.12), entonces su total de energía térmica es de 1.325 BTU/h. Energía térmica total producida anual es de 11607 BTU/h equivalentes a 3400 Watts.

**Escenario sin Virtualizar**

En este escenario se tienen 3 servidores (Figura 5.11) cada servidor genera 1.325 BTU/h y con ello se tiene un total de energía térmica de 3.975 BTU/h. Energía térmica total producida anual es de 34821 BTU/h equivalentes a 1200 Watts

**Análisis de Resultados**



Grafica 5.12 Consumo anual energía térmica

En la gráfica 5.12 se muestra que la energía térmica producida en un ambiente sin virtualizar es mucho mayor que uno virtualizado, es 3 veces mayor y esto se genera una mayor cantidad de calor, entonces para mantener el Network Operations Center (NOC) en

este escenario a su temperatura de operación óptima se requiere de equipos de mayor capacidad de enfriamiento y mejor infraestructura lo que da lugar a incrementos en costos de energía.

#### 5.4. Pruebas de Funcionalidad

Se realizaron pruebas de funcionalidad para corroborar que la Arquitectura de Virtualización con Solaris Containers cumpla con las características especificadas en el Capítulo 3 de Diseño de la Arquitectura.

##### - Verificación de redundancia en los discos de arranque del Sistema Operativo

**Objetivo:** Verificar la configuración y funcionalidad de la redundancia de los discos de arranque

**Resultado esperado:** El sistema debe iniciar desde el disco primario y espejo sin problemas. Los discos deben sincronizarse sin problema ante cualquier eventualidad.

##### Procedimiento:

1. Verificar que tanto el disco de arranque como su disco espejo tengan tabla de partición idéntica.

Tabla de partición disco de arranque:

```
c1t0d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 6183	60.01GB	(6184/0/0) 125856768
1	swap	wu	6184 - 7832	16.00GB	(1649/0/0) 33560448
2	backup	wm	0 - 14086	136.71GB	(14087/0/0) 286698624
3	unassigned	wm	7833 - 8657	8.01GB	(825/0/0) 16790400
4	unassigned	wm	8658 - 9482	8.01GB	(825/0/0) 16790400
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wm	9483 - 14060	44.43GB	(4578/0/0) 93171456
7	unassigned	wm	14061 - 14086	258.38MB	(26/0/0) 529152

Tabla de partición disco espejo:

```
c1t1d0 <LSILOGIC-LogicalVolume-3000 cyl 65533 alt 2 hd 16 sec 273>
```



Current partition table (original):

Total disk cylinders available: 65533 + 2 (reserved cylinders)

Part	Tag	Flag	Cylinders	Size	Blocks
0	root	wm	0 - 28855	60.10GB	(28856/0/0) 126043008
1	unassigned	wu	28856 - 36539	16.00GB	(7684/0/0) 33563712
2	backup	wu	0 - 65532	136.49GB	(65533/0/0) 286248144
3	unassigned	wm	36540 - 40385	8.01GB	(3846/0/0) 16799328
4	unassigned	wm	40386 - 44241	8.01GB	(3856/0/0) 16843008
5	unassigned	wm	0	0	(0/0/0) 0
6	unassigned	wm	44242 - 65411	44.43GB	(21170/0/0) 92470560
7	unassigned	wm	65412 - 65532	258.07MB	(121/0/0) 528528

## 2. Verificar la existencia de la base de metadispositivos y sus réplicas.

```
uxsop001 / # metadb -i
```

```

  flags          first blk      block count
  a m p lu0     16             8192        /dev/dsk/clt0d0s7
  a  p lu0     8208             8192        /dev/dsk/clt0d0s7
  a  p lu0    16400             8192        /dev/dsk/clt0d0s7
  a  p lu0     16             8192        /dev/dsk/clt1d0s7
  a  p lu0     8208             8192        /dev/dsk/clt1d0s7
  a  p lu0    16400             8192        /dev/dsk/clt1d0s7
r - replica does not have device relocation information
o - replica active prior to last mddb configuration change
u - replica is up to date
l - locator for this replica was read successfully
c - replica's location was in /etc/lvm/mddb.cf
p - replica's location was patched in kernel
m - replica is master, this is replica selected as input
W - replica has device write errors
a - replica is active, commits are occurring to this replica
M - replica had problem with master blocks
D - replica had problem with data blocks
F - replica had format problems
S - replica is too small to hold current data base
R - replica had device read errors

```

3. Ejecutar el comando `metastat -p` y verificar que las características de los metadispositivos listado corresponda con las particiones y dispositivos en uso.

```

uxsop001 / # df -h
Filesystem                size      used  avail capacity  Mounted on
/dev/md/dsk/d0             59G       22G   37G    37%      /
/devices                   0K         0K    0K     0%      /devices
ctfs                       0K         0K    0K     0%      /system/contract
proc                       0K         0K    0K     0%      /proc
mnttab                     0K         0K    0K     0%      /etc/mnttab
swap                       20G       1.7M   20G     1%      /etc/svc/volatile
objfs                      0K         0K    0K     0%      /system/object
sharefs                    0K         0K    0K     0%      /etc/dfs/sharetab
/platform/SUNW,Sun-Fire-T200/lib/libc_psr/libc_psr_hwcap1.so.1
                                                                    37%
                                                                    59G          22G
/platform/sun4v/lib/libc_psr.so.1
/platform/SUNW,Sun-Fire-T200/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1
                                                                    37%
                                                                    59G          22G
/platform/sun4v/lib/sparcv9/libc_psr.so.1
fd                          0K         0K    0K     0%      /dev/fd
swap                        20G       112K   20G     1%      /tmp
swap                        20G        64K   20G     1%      /var/run
/dev/md/dsk/d4             7.9G      8.0M   7.8G     1%      /var/crash
zones                       67G       29K   37G     1%      /zones
zones/colaboracion-app     10G       5.0G   5.0G    51%     /zones/colaboracion-app
zones/colaboracion-db     10G       5.0G   5.0G    51%     /zones/colaboracion-db
zones/ecommerce-app       10G       5.0G   5.0G    51%     /zones/ecommerce-app
zones/ecommerce-db       10G       5.0G   5.0G    51%     /zones/ecommerce-db
zones/joomla-app          10G       5.0G   5.0G    51%     /zones/joomla-app
zones/joomla-db           10G       5.0G   5.0G    51%     /zones/joomla-db

```

```

uxsop001 / # dumpadm -d swap
Dump content: kernel pages
Dump device: /dev/md/dsk/d1 (swap)
Savecore directory: /var/crash/uxsop001
Savecore enabled: yes
Save compressed: on

uxsop001 / # metastat -p
d4 -m d14 d24 1
d14 1 1 c1t0d0s4
d24 1 1 c1t1d0s4
d1 -m d11 d21 1
d11 1 1 c1t0d0s1
d21 1 1 c1t1d0s1
d0 -m d10 d20 1
d10 1 1 c1t0d0s0
d20 1 1 c1t1d0s0

```

4. Verificar que el archivo `/etc/system` contenga la variable `rootdev`, cuyo valor debe ser la ruta del metadispositivo espejo que contiene al sistema de archivos `root`.

```

uxsop001 / # cat /etc/system | grep root
* root device and root filesystem configuration:
*   rootfs:          Set the filesystem type of the root.
*   rootdev:        Set the root device.  This should be a fully
*                   rootfs:ufs
*                   rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
*   the root filesystem) rather than at first reference. Note that
* Begin MDD root info (do not edit)
rootdev:/pseudo/md@0:0,0,blk

```

5. Verificar que el archivo `/etc/vfstab` contenga todos los metadispositivos configurados en los discos de arranque

```

uxsop001 / # cat /etc/vfstab
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type     pass     at boot   options
#
fd           -          /dev/fd fd    -        no        -
/proc       -          /proc  proc  -        no        -
/dev/md/dsk/d1 -          -        swap  -        no        -
/dev/md/dsk/d0 /dev/md/rdisk/d0 /        ufs     1        no        -
/dev/md/dsk/d4 /dev/md/rdisk/d4 /var/crash ufs     2        yes       -
/devices    -          /devices devfs  -        no        -
sharefs     -          /etc/dfs/sharetab sharefs -        no        -
ctfs        -          /system/contract ctfs   -        no        -
objfs       -          /system/object objfs  -        no        -
swap        -          /tmp    tmpfs  -        yes       -

```

6. Cambiar el estado del servidor a OBP. Verificar la existencia de un alias hacia el disco de arranque y un alias hacia el disco espejo del sistema.

```

uxsop001 / # sync; sync; shutdown -i0 -g0 -y
Shutdown started.      Sat Nov 27 14:59:32 CST 2010
Changing to init state 0 - please wait
Broadcast Message from root (console) on uxsop001 Sat Nov 27 14:59:32...
THE SYSTEM uxsop001 IS BEING SHUT DOWN NOW !!!
Log off now or risk your files being damaged
showmount: uxsop001: RPC: Program not registered
uxsop001 / # svc.startd: The system is coming down. Please wait.
svc.startd: 93 system services are now being stopped.
Nov 27 14:59:48 uxsop001 rpc.metad: Terminated
Nov 27 14:59:56 uxsop001 syslogd: going down on signal 15
svc.startd: The system is down.
syncing file systems... done
Program terminated
{7} ok
{7} ok devalias
disk /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@0
diskmirror /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@1
ttya /pci@7c0/pci@0/pci@1/pci@0/isa@2/serial@0,3f8
nvram /virtual-devices/nvram@3
net3 /pci@7c0/pci@0/pci@2/network@0,1
net2 /pci@7c0/pci@0/pci@2/network@0
net1 /pci@780/pci@0/pci@1/network@0,1
net0 /pci@780/pci@0/pci@1/network@0
net /pci@780/pci@0/pci@1/network@0
ide /pci@7c0/pci@0/pci@1/pci@0/ide@8
cdrom /pci@7c0/pci@0/pci@1/pci@0/ide@8/cdrom@0,0:f
scsi /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
virtual-console /virtual-devices/console@1
{7} ok printenv boot-device
boot-device = disk diskmirror

```

## 7. Iniciar el Sistema Operativo desde el disco espejo.

```

{7} ok boot diskmirror
SC Alert: Host System has Reset
_ChassisSerialNumber 0540NNN092
Sun Fire T200, No Keyboard
Copyright 2010 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.30.4.a, 8064 MB memory available, Serial #64551886.
Ethernet address 0:3:ba:d8:fb:ce, Host ID: 83d8fbce.

Boot device: /pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/disk@1 File and
args:
SunOS Release 5.10 Version Generic_142909-17 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights
reserved.
Hostname: uxsop001
/dev/md/rdisk/d4 is clean
Reading ZFS config: done.
Mounting ZFS filesystems: (7/7)
uxsop001 console login: root
Password:
Nov 27 15:12:59 uxsop001 login: ROOT LOGIN /dev/console
Last login: Sat Nov 27 14:21:29 from 192.168.82.126
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
uxsop001 / # uptime
 3:13pm up 1 min(s), 1 user, load average: 2.78, 0.80, 0.28
uxsop001 / # uname -a
SunOS uxsop001 5.10 Generic_142909-17 sun4v sparc SUNW,Sun-Fire-T200

```

8. Reiniciar e iniciar el Sistema Operativo en modo normal.

```

uxsop001 / # sync; sync; shutdown -g0 -i0 -y
Shutdown started.      Sat Nov 27 15:15:34 CST 2010
Changing to init state 0 - please wait
Broadcast Message from root (console) on uxsop001 Sat Nov 27 15:15:34...
THE SYSTEM uxsop001 IS BEING SHUT DOWN NOW ! ! !
Log off now or risk your files being damaged
uxsop001 / # svc.startd: The system is coming down.  Please wait.
svc.startd: 93 system services are now being stopped.
Nov 27 15:16:39 uxsop001 rpc.metad: Terminated
Nov 27 15:16:40 uxsop001 syslogd: going down on signal 15
svc.startd: The system is down.
syncing file systems... done
Program terminated
{3} ok boot

SC Alert: Host System has Reset
_ChassisSerialNumber 0540NNN092

Sun Fire T200, No Keyboard
Copyright 2010 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.30.4.a, 8064 MB memory available, Serial #64551886.
Ethernet address 0:3:ba:d8:fb:ce, Host ID: 83d8fbce.

Boot device: disk File and args:
SunOS Release 5.10 Version Generic_142909-17 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates.  All rights
reserved.
Hostname: uxsop001
/dev/md/rdisk/d4 is clean
Reading ZFS config: done.
Mounting ZFS filesystems: (7/7)
uxsop001 console login: root
Password:
Last login: Sat Nov 27 15:12:59 on console
Nov 27 15:20:35 uxsop001 login: ROOT LOGIN /dev/console
Sun Microsystems Inc.      SunOS 5.10      Generic January 2005
uxsop001 / # uptime
  3:20pm  up 1 min(s),  1 user,  load average: 3.53, 1.09, 0.39

```

9. Retirar físicamente el disco espejo y volverlo a insertar. Validar que una vez insertado el disco espejo se inicie el proceso de resincronización y que el sistema no sufrió afectación alguna.

- **Verificación de redundancia en la red**

**Objetivo:** Validar la funcionalidad de la configuración de IPMP

**Resultado esperado:** Al detectarse una falla en cualquiera de las interfaces de red no debe presentarse intermitencia en la red.

**Procedimiento:**

1. Verificar que las dos tarjetas de red se encuentren funcionando.

```

uxsop001 / # ifconfig -a
e1000g0:
flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 2
    inet 192.9.198.201 netmask fffffff0 broadcast 192.9.198.255
    groupname ipmp0
    ether 0:3:ba:d8:fb:ce
e1000g0:1:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    inet 192.9.198.190 netmask fffffff0 broadcast 192.9.198.255
e1000g0:2:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    zone colaboracion-db
    inet 192.9.198.91 netmask fffffff0 broadcast 192.9.198.255
e1000g0:3:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    zone colaboracion-app
    inet 192.9.198.185 netmask fffffff0 broadcast 192.9.198.255
e1000g0:4:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    zone ecommerce-db
    inet 192.9.198.71 netmask fffffff0 broadcast 192.9.198.255
e1000g0:5:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    zone ecommerce-app
    inet 192.9.198.235 netmask fffffff0 broadcast 192.9.198.255
e1000g0:6:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    zone joomla-app
    inet 192.9.198.213 netmask fffffff0 broadcast 192.9.198.255
e1000g0:7:  flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>  mtu  1500
index 2
    zone joomla-db
    inet 192.9.198.212 netmask fffffff0 broadcast 192.9.198.255
e1000g2:
flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
    inet 192.9.198.130 netmask fffffff0 broadcast 192.9.198.255
    groupname ipmp0
    ether 0:3:ba:d8:fb:d0

```

2. Ejecutar el comando ping hacia el servidor.(Figura 5.13)

```

C:\WINDOWS\system32\cmd.exe
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Reply from 192.9.198.190: bytes=32 time<1ms TTL=251
Ping statistics for 192.9.198.190:
    Packets: Sent = 541, Received = 540, Lost = 1 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
U:\>
    
```

Figura 5.13 Ejecución de comando ping

3. Desconectar físicamente el cable de red de la tarjeta primaria y validar que se realice el cambio de las direcciones IP hacia la tarjeta secundaria.

```

uxsop001 /var/adm # dmesg
Nov 19 17:04:32 uxsop001 e1000g: [ID 801725 kern.info] NOTICE: pciex8086,105e -
e1000g[0] : link down
Nov 19 17:04:32 uxsop001 in.mpathd[178]: [ID 215189 daemon.error] The link has
gone down on e1000g0
Nov 19 17:04:32 uxsop001 in.mpathd[178]: [ID 594170 daemon.error] NIC failure
detected on e1000g0 of group ipmp0
Nov 19 17:04:32 uxsop001 in.mpathd[178]: [ID 832587 daemon.error] Successfully
failed over from NIC e1000g0 to NIC e1000g2
uxsop001 / # ifconfig -a
e1000g0:
flags=19040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER,FAILED>
mtu 1500 index 2
    inet 192.9.198.201 netmask ffffffff broadcast 192.9.198.255
    groupname ipmp0
    ether 0:3:ba:d8:fb:ce
e1000g2: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3
    inet 192.9.198.130 netmask ffffffff broadcast 192.9.198.255
    groupname ipmp0
    ether 0:3:ba:d8:fb:d0
e1000g2:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 192.9.198.190 netmask ffffffff broadcast 192.9.198.255
e1000g2:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    zone colaboracion-db
    inet 192.9.198.91 netmask ffffffff broadcast 192.9.198.255
e1000g2:3: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    zone colaboracion-app
    inet 192.9.198.185 netmask ffffffff broadcast 192.9.198.255
e1000g2:4: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    zone ecommerce-db
    inet 192.9.198.71 netmask ffffffff broadcast 192.9.198.255
e1000g2:5: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    zone ecommerce-app
    inet 192.9.198.235 netmask ffffffff broadcast 192.9.198.255
e1000g2:6: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    zone joomla-app
    inet 192.9.198.213 netmask ffffffff broadcast 192.9.198.255
e1000g2:7: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    zone joomla-db
    inet 192.9.198.212 netmask ffffffff broadcast 192.9.198.255
    
```

4. Conectar físicamente el cable de red de la tarjeta primaria y validar su incorporación al grupo de IPMP.

```

uxsop001 /var/adm # dmesg
Nov 19 17:08:07 uxsop001 in.mpathd[178]: [ID 820239 daemon.error] The link
has come up on e1000g0
Nov 19 17:08:52 uxsop001 in.mpathd[178]: [ID 299542 daemon.error] NIC
repair detected on e1000g0 of group ipmp0
Nov 19 17:08:52 uxsop001 in.mpathd[178]: [ID 620804 daemon.error]
Successfully failed back to NIC e1000g0

```

5. Desconectar físicamente el cable de red de la tarjeta secundaria, conectar nuevamente y validar su incorporación al grupo de IPMP.

```

uxsop001 /var/adm # dmesg
Nov 19 17:09:54 uxsop001 in.mpathd[178]: [ID 215189 daemon.error] The link
has gone down on e1000g2
Nov 19 17:09:54 uxsop001 in.mpathd[178]: [ID 594170 daemon.error] NIC
failure detected on e1000g2 of group ipmp0
Nov 19 17:09:54 uxsop001 in.mpathd[178]: [ID 832587 daemon.error]
Successfully failed over from NIC e1000g2 to NIC e1000g0
Nov 19 17:10:31 uxsop001 e1000g: [ID 801725 kern.info] NOTICE:
pciex8086,105e - e1000g[2] : link up, 100 Mbps, full duplex
Nov 19 17:10:31 uxsop001 in.mpathd[178]: [ID 820239 daemon.error] The link
has come up on e1000g2

```



# Conclusiones

---

*En esta sección analizamos los resultados obtenidos a lo largo de la realización de la Tesis. Resaltando los objetivos y las aportaciones.*

La virtualización de servidores es una técnica que se utiliza para integrar varios servicios o aplicaciones (servidores virtuales) en un solo servidor físico. Bajo este concepto se diseñó e implementó una arquitectura de servidores virtualizados usando la técnica de Solaris Containers, para optimizar los recursos de hardware del servidor SunFire T2000, que es el objetivo de este trabajo, el cual se ha cumplido satisfactoriamente y se ha podido evaluar el sistema virtualizado a través de varias pruebas.

Para la virtualización de servidores, se han tomado en consideración los beneficios de su uso, las distintas alternativas existentes en la industria para su desarrollo, los elementos con que interactúan, bajo la experiencia de los redactores. Es importante mencionar que gran parte de la información está sustentada por fuentes diversas, desde libros, documentación pública de los fabricantes de TI, hojas de datos de dispositivos y estudios hechos por consultoras del ramo.

Las características más importantes que se destacan de este trabajo son:

- Está integrado como una guía
- Se incluye una Tabla Comparativa de las Diferentes Técnicas de Virtualización
- Aporta elementos de Evaluación
- Se incluye una Línea de la Evolución de la virtualización
- Se incluye una Tabla de Metodologías

Una aportación de este trabajo es la recopilación de múltiples fuentes de datos sobre la evolución de la virtualización. Se incluyó una línea de tiempo con los principales acontecimientos, la cual muestra cómo han sido los esfuerzos en la investigación y desarrollo por parte de los proveedores de tecnologías. En ella se observa que en los últimos 10 años han surgido la mayoría de las soluciones comerciales actuales, haciendo de la virtualización un tema en boga.

Aún cuando el tema de virtualización no es nuevo, no se le da la difusión que amerita y ha tomado mayor importancia y lugar en las organizaciones desde el año 2005. En ese sentido, este trabajo describe el tema y además consideramos que sirve como una guía técnica de diseño e implementación y aporta los elementos para una evaluación completa del sistema, haciendo de éste un documento integral.

La Tabla Comparativa de las Diferentes Técnicas de Virtualización y la Línea de la Evolución de la Virtualización no se encuentran en otro documento, siendo por tanto resultado de la investigación que se llevó a cabo para este trabajo, las cuales consideramos que son aportaciones importantes de esta Tesis.

Si bien, existe literatura que habla de los distintos productos de virtualización en el mercado, estos documentos están enfocados a un producto en particular y en ocasiones

se omite información. El esfuerzo de recopilar toda la información sobre los tipos de virtualización y los productos existentes se ve reflejado en la tabla de Comparación de las Técnicas de Virtualización que concentra las características de cada una de las técnicas.

Al ser un documento integral, presenta varias facetas de la virtualización, desde los componentes técnicos necesarios para llevar a cabo un escenario de servidores virtualizados, hasta los argumentos financieros y administrativos que hacen de la virtualización una solución a diversos problemas que se presentan en los entornos de cómputo de una organización actual, además para dar formalidad al sistema desarrollado, se sigue una metodología general de proyectos.

El tema de la virtualización de servidores se enfoca, en este trabajo, en la tecnología de Solaris Containers o Contenedores ya que ésta cuenta con características especiales que resalta sobre las demás. Solaris Containers es una tecnología sin costo de licenciamiento, es de distribución libre, permite virtualizar ambientes Linux y UNIX, puede implementarse en plataformas x86, x64 y SPARC y se pueden virtualizar ambientes de Solaris 8 y 9 dentro de Solaris 10.

Los resultados que obtuvimos con la realización de esta Tesis indican que se consigue un ahorro de energía en el centro de datos, esto debido a que se disminuye el equipo de hardware al virtualizar servidores. De tener 3 servidores funcionando y consumiendo energía, con la virtualización pasamos a tener únicamente 1 servidor, el cual es capaz de entregar los mismos servicios que ofrecían originalmente los primeros 3 servidores.

También pudimos comprobar a través del análisis de TCO que una solución de servidores virtualizados resultan no sólo en ahorro de energía, también ayudan a disminuir los costos de todo lo relacionado a un centro de datos, como por ejemplo costos de administración, costos de comunicaciones, costos de espacio, costos de capacitación, costos de actualización, etc.

La virtualización de servidores es una solución a muchos problemas que enfrentan actualmente los centros de datos y las organizaciones; pero también debe de considerarse que la virtualización puede no ser ideal para todas las aplicaciones o servicios, esto debido a diversas situaciones, entre las que se encuentran Software no virtualizable, Software de monitoreo y seguridad, entre otras. Aplicaciones críticas o servicios críticos, pueden requerir de todos los recursos de hardware disponibles y sin

compartir con otros, lo que probablemente implicaría contar con un servidor físico dedicado.

Otra de las ventajas de usar Solaris Containers para virtualizar servidores, es que los recursos totales de hardware pueden asignarse exclusivamente a un servicio o aplicación en particular, evitando que se comparta con los demás servidores virtualizados que existan en el servidor físico, lo cual realizamos con éxito en esta Tesis.

Consideramos que esta Tesis puede servir como referencia en un proyecto de virtualización de servidores, ya que presenta una metodología del diseño y arquitectura de la solución, el análisis antes y después de la implementación de virtualización y también proporciona un análisis de costo para evaluar los beneficios de la implementación de virtualización de servidores. Independientemente de la herramienta que se use para la virtualización de servidores, este trabajo aporta información necesaria, como: datos, metodología, análisis y pruebas que se recomiendan como “mejores prácticas” para llevar a cabo un proyecto de ésta índole.

# Glosario de términos

---

**Arquitectura:** diseño específico de los componentes de un sistema y la forma en que interactúan con los demás.

**Capacidad de Servicio (Serviceability) :** es la cualidad que está directamente relacionada con el buen nivel la fiabilidad y disponibilidad que se ofrezca, de tal forma que se pueda identificar que una manera muy rápida un problema o falla para obtener la solución lo antes posible, evitando así tiempos muertos (system downtime).

**Clúster:** es un grupo de dos o más servidores independientes que trabajan juntos como un sistema único para proporcionar alta disponibilidad, un mayor nivel de confiabilidad y escalabilidad, están diseñados para proteger los datos, mantener las aplicaciones y servicios.

**Disponibilidad (Availability):** es la cualidad de mantener y asegurar que el servicio o sistema sea capaz de mantenerse activo y en funcionamiento realizando las funciones para las que está diseñado, de tal forma que se reduzcan los tiempos de inactividad.

**Escalabilidad:** es la capacidad que tiene un servicio, una aplicación y un sistema de crecer, con la finalidad de satisfacer demandas de rendimiento que se van incrementando. En el escalado vertical se incorporan nuevos recursos del sistema, procesadores, memoria, discos y adaptadores de red, mientras que en el horizontal se agregan nuevos servidores.

**Fiabilidad (Reliability):** es la medida de tolerancia que presenta un sistema en cuanto a errores, de tal forma que se pueda seguir funcionando correctamente aunque se presenten estos, para obtener un grado mayor se debe diseñar el sistema con un alto grado de redundancia. Es crucial en los sistemas modernos, incluso si esta incorporación pueda alentar el sistema, pero es preferible eso a que se presenten fallas y pérdida de datos,

**Hypervisor o Virtual Machine Monitor (VMM):** es una plataforma de virtualización que permite ejecutar varios sistemas operativos en un solo equipo físico, su principal función es el de proveer ambientes de ejecución aislados para cada máquina virtual, administrarlas y manejar los accesos entre el sistema operativo guest y los subyacente recursos. Los hypervisor´s se pueden clasificar de acuerdo a su diseño y a su tipo, en el tipo 1 se ejecuta directamente en la capa hardware física subyacente es por esto que se le conoce como unhosted o sobre el metal desnudo (bare metal), funciona como un

programa de control y las máquinas virtuales se ejecutan en una capa arriba de este Microsoft HyperV. XenServer, y VMware ESXi Server implementan este VMM, mientras que el tipo 2 se ejecuta sobre el sistema operativo host.

**Infraestructura tecnológica.-** Conjunto de dispositivos y tecnologías que se refieren generalmente a elementos físicos

**Máquina Virtual (VM):** es un contenedor de software en ambiente aislado el cual abstrae los recursos físicos del hardware, ejecuta sus propios sistemas operativos y aplicaciones comportándose exactamente igual que lo hace un equipo físico y contiene sus propios recursos CPU, disco duro, memoria, y tarjetas de interfaz de red.

**Modelo en capas:** es un método utilizado en Ingeniería de software, para dividir la aplicación en n capas. Cuando  $n = 3$ , tenemos un modelo de 3 capas, donde cada nivel tiene un tipo de abstracción diferente al de los demás, aunque cada capa tiene comunicación con las demás capas.

Capa de Datos: Esta capa es la encargada de abstraer los datos a utilizar en la aplicación.

Capa de Aplicación o Lógica del Negocio: Esta capa se encarga de la intercomunicación entre la capa de datos, con la capa de presentación.

Capa de Presentación: Es la capa donde se presentan los datos al usuario, puede ser a través de html, archivos de texto, etc.

**Nivel de servicio.-** Acuerdos entre los proveedores de servicios y los usuarios para determinar aspectos como tiempo de respuesta, disponibilidad, tiempo para recuperación en caso de desastre, tiempo fuera del servicio entre otros

**RAS (Reliability Availability Serviceability).** Es un conjunto de cualidades que se aplican en el diseño y/o implementación de servicios, aplicaciones y equipos como un estándar de calidad.

**RAID (Redundant Array of Independent Disks):** es un arreglo de discos con varios niveles, cada uno de los cuales aplica distintas técnicas para ofrecer tolerancia a fallas en el subsistema de discos del servidor, existen arreglos del RAID 0 hasta el RAID 6, además de que existen RAID híbridos que son combinaciones de técnicas como RAID 1+0.

RAID 0: En este nivel la información se divide entre todos los discos que componen el arreglo, distribuyéndolos entre los discos. De todos los niveles es el de menor costo, aunque también el que menor seguridad ofrece.

RAID 1 (Mirror): Por cada disco en el arreglo se necesita otro de la misma capacidad para duplicar o hacer el “espejo” de la información que se almacena. Si uno de los discos falla, la información se encontrará almacenada en el otro. Se tiene una alta disponibilidad de los datos, pero a un costo muy elevado debido a que se requiere el doble de discos y se cuenta con sólo la mitad de la capacidad total.

**RAID 5:** los datos en se dividen en todos los discos del arreglo al nivel de bloque o sector, pero junto con ellos la información de la paridad, sin que para esta tarea se dedique un disco en particular como en el RAID 3 y 4. De esta forma se solucionan los inconvenientes en desempeño al tener un solo disco de paridad, este arreglo se sugiere en sistemas multiusuario, donde el desempeño no es un factor crítico, pero se requiere confiabilidad.

**Redundancia:** es la incorporación de más recursos al sistema, de tal forma que no se tenga un único recurso destinado a una tarea específica, por lo que si falla un recurso se tenga otro igual características que realice sus misma funciones, y así se evita tener un único punto de falla.

Los componentes redundantes comunes en un servidor suelen ser, los discos, las tarjetas de red y las fuentes de alimentación.

**Rendimiento:** es la medida de aprovechamiento de un recurso, aplicación o un equipo, siempre se está en busca el máximo en términos de eficacia y de eficiencia.

**Servidor.-** Físicamente, es un equipo de cómputo con mayor cantidad de recursos que un equipo de escritorio. Lógicamente, es un equipo o conjunto de equipos de cómputo que proveen servicios a través de la red

**Sistema Operativo:** es un conjunto de programas que se integran con el hardware, en general provee de un ambiente conveniente de trabajo, asigna los recursos y maneja una adecuada distribución de estos, al igual que administra y controlar la ejecución de los programas.

**Sistema Operativo Host:** es el sistema operativo que se le conoce como “base” o “anfitrión”, se ejecuta en el hardware de virtualización, tiene acceso directo al hardware subyacente y, en general alberga las máquinas virtuales.

**Sistema Operativo Guest:** es el sistema operativo que se instala y ejecuta en las máquinas virtuales se le conoce como “invitado”, permite la ejecución de aplicaciones que se ejecutan e interactúan con el ambiente virtual como si estuvieran en un equipo dedicado.

# Referencias bibliográficas

---

CLARK, "Storage Virtualization: Technologies for Simplifying Data Storage and Management", Addison Wesley Press, 2005,

DESAI, "The Definitive Guide To Virtual Platform Management", Real Time Publishers, USA, 2007.

DETLEF, ULRICH, et al, "Solaris 10 Container Guide", Sun Microsystems Inc, 2009.

FERGUSON, HUSTON, "What Is a VPN?", Cisco Systems, Tech. U.S.A., 1998.

GOLDEN, SCHEFFY, "Virtualization for Dummies, Sun and AMD Special Edition", Wiley Publishing, 2008.

HP Inc., "Guía del usuario de HP Integrity Essentials, Capacity Advisor", HP Press, 2006

HEALEY, MATT, et all, "White Paper, IBM Virtualization Services", <http://www-935.ibm.com/services/us/its/html/idc-virtualization.html>, publicacado en Octubre 2008, última consulta Septiembre 2010.

HESS, NEWMAN, "Practical Virtualization Solutions: Virtualization from the Trenches", Pearson Education Inc., USA 2010

HOMBY, PEPPLE, "Consolidation in the Data Center, Simplifying IT environments to reduce Total Cost of Ownership", Sun Blueprints, Prentice Hall, 2003

IBM Corp, "Redbooks Paper: VMware ESX Server: Scale Up or Scale Out?", California, U.S.A, 2004.

McDOUGALL, MAURO, "Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture, Second Edition", Prentice Hall, 2006.

Microsoft Inc., "Virtualization Solutions: From the Desktop to the Datacenter", Microsoft Press. Second Edition, Washington, USA, 2010.

NASCIO, "Enterprise Data Center Consolidation in the States: Strategies and Business Justification", 2007.



Oracle Inc, "Oracle White Paper: Consolidating Applications with Oracle Solaris Containers", 2010.

PONIATOWSKI, "Foundation of Green IT", Agosto 2009.

RAY, RAY, "Visual QuickStart Guide Unix and Linux, Fourth Edition", Peachpit Press, 2009.

ROSENBLUM, GARFINKEL, "Virtual machine monitors: current technology and future trends", IEEE Computer Society Press, Los Alamitos, U.S.A, 2005.

SHIELDS, "Important Questions in Implementing Virtual Desktops", Real Time Publishers, USA, 2009.

Sun Microsystems Inc., "System Administrator Guide: Virtualization Using the Solaris Operating System", California, U.S.A., 2009.

Sun Microsystems Inc, "White Paper Solaris Containers: Server Virtualization and Manageability", California, U.S.A., 2004

Sun Microsystems Inc., "Solaris 10 system administration essentials", California, U.S.A., 2010.

VMware Inc., "VMware White Paper: Best Practices Using Virtual SMP", San José, California, U.S.A, 2005.

"Antecedentes", <http://www-03.ibm.com/ibm/history/>

"Antecedentes", <http://www.vmware.com/es/virtualization/history.html>

"Antecedentes", <http://return42.blogspot.com/2009/05/how-ibm-started-virtual-revolution.html>

"Virtualización de Aplicaciones", <http://download.microsoft.com/download/6/4/f/64f5dc66-832a-4df3-baf4-3b4e7fb9e500/SGAV%20White%20Paper%20-The%20Next%20Frontier.pdf>

"Kernel Virtual Machine", <http://kvm.qumranet.com/>

"Kernel Virtual Machine", <http://www.ibm.com/developerworks/linux/library/l-linux-kvm/>

"Qué es la virtualización?", <http://www.vmware.com/es/virtualization/what-is-virtualization.html>

"VMWare ESXi", <http://www.wetcom.com.ar/content/vmware-lanzo-una-herramienta-de-administracion-centralizada-para-esxi/>

“Virtualización: concepto”, <http://www.wetcom.com.ar/servicios/virtualizacion/>

“VMware ESXi”, <http://www.wetcom.com.ar/content/derribando-algunos-mitos-de-vmware-esxi/>

“HP Virtualización”, <http://docs.hp.com/es/T2784-90019/ch05s02.html>

“VMware ESX Server”, <http://www.wetcom.com.ar/content/tag/esx/>

“Ventajas y Desventajas de la Virtualización”, <http://www.techweek.es/virtualizacion/tech-labs/1003109005901/ventajas-desventajas-virtualizacion.2.html>

“Análisis Costo Total de Propiedad”,  
[http://www.gartner.com/4\\_decision\\_tools/measurement/decision\\_tools/tco/tco\\_adc.html](http://www.gartner.com/4_decision_tools/measurement/decision_tools/tco/tco_adc.html)

# Anexos

---

*Se documenta de forma extensa procesos de instalación, configuración y puesta a punto del sistema virtualizado. Además se incluye información y documentos adicionales.*

## ANEXO 1 – Hoja de Datos SunFire T2000

Una Hoja de Datos, también conocido como Datasheet, es un documento que contiene un resumen de las características y beneficios de un producto o servicio. A continuación se muestra la hoja de datos del servidor usado en este trabajo de tesis.



### Highlights

- World's first eco-responsible server
- Up to 32 simultaneous execution threads using CoolThreads™ chip multithreading (CMT) technology
- Industry-leading server efficiency as defined by space, watts, and performance (SWaP) metric
- Drive up utilization, drive down costs, and massively consolidate workloads with built-in, no-cost virtualization technology with Logical Domains (LDoms) and Solaris™ Containers
- Investment protection with SPARC® V9 binary application compatibility
- Enhanced system uptime supported by high on-chip RAS, redundant power supplies and fans, hot-pluggable disk drives, hardware RAID (0+1), and low component count
- Exceptional expandability and I/O performance from five PCI slots and four Ethernet interfaces
- Preloaded with the Solaris 10 OS and CoolTools for CoolThreads server to enable fast application development and deployment
- Runs many Linux distributions, so you can choose the best operating system for your environment



With the growing demand for network services, datacenter managers are caught between competing requirements: building server capacity and deploying applications quickly on the one hand, while managing complexity and server sprawl and coping with burgeoning energy costs and real estate limitations on the other. So companies are looking for systems that can deliver on all the requirements of the next-generation datacenter—for compute density, for price/performance, and for energy and space efficiency. That's the whole idea behind the Sun Fire™ T2000 server.

The Sun Fire T2000 server is powered by the UltraSPARC® T1 processor, which delivers four or eight cores for a maximum of 32 simultaneous software threads per chip. The result? Up to five times the throughput-optimized performance of competitors' systems—performance that's ideal not only for Web-tier applications, but for critical OLTP databases and ERP, CRM, and SCM applications as well.

For these types of applications, your systems have to be up, online, and available. And the Sun Fire T2000 server delivers enhanced uptime capabilities you can rely on: hot-pluggable disk drives, redundant and hot-swappable power supplies and fans, environmental monitoring, RAID 0+1 monitoring, and much more. Plus the component count itself is extremely low—and fewer components mean better reliability and less system downtime.

Sun delivers all this performance and reliability in a low-profile, 2U design ideal for rackmount deployments, while it draws about one-fifth the power of other systems. The combination of high performance and RAS features with minimal energy costs and physical footprint is so innovative that we had to develop a new space, watts, and power (SWaP) metric to describe its true cost and environmental impact on the datacenter.

Plus it comes preinstalled with the highly available, secure, reliable Solaris 10 OS. With the Solaris Application Guarantee, we ensure binary compatibility between Solaris OS releases. Plus, Logical Domains (LDoms) and Solaris Containers enable you to safely consolidate dozens of applications onto a single system to virtualize your environment, increase utilization rates, and cut system and licensing costs.

## Sun Fire T2000 Server Specifications

### Architecture

#### Processor

Four- or eight-core 1.0 GHz, or eight-core 1.2 GHz UltraSPARC T1 processor

#### Cache

- 16 KB instruction processor
- 8 KB primary data cache
- 3 MB integrated L2

#### Main memory

16 DIMM slots, DDR2 with Chipkill and DRAM sparing, ECC Registered DIMMs, system maximum of 64 GB; support for 1 GB, 2 GB, and 4 GB DIMMs

#### KeyIO features

- Hot-pluggable disk drives
- Redundant, hot-swappable power supplies
- Redundant, hot-swappable fans
- Environmental monitoring
- Error correction and parity checking
- Easy component replacement
- Internal hardware drive mirroring (RAID 1)
- RAID 0 and 1 support

### System architecture

SPARC V9 architecture, ECC protected

### Standard integrated interfaces

- Network: Four 10/100/1000 Mb/sec Ethernet
- Serial management: One DB-9 serial port
- USB: Four 2.0 USB ports
- Expansion bus: Three PCI Express (PCIe) slots for low-profile cards (support x1, x4 and x8 width cards); two PCI-X slots for 64-bit 133 MHz low-profile cards (Note: On earlier models, one PCI-X slot might be occupied with a disk controller card)

### Software

#### Operating system

- Solaris 10 OS 11/06

#### Pre-installed software

- Solaris 10
- Cool Tools for CoolThreads servers (includes CoolTuner, Sun Studio 11, GCC for SPARC system)

### Management

- Advanced Lights Out Manager (ALOM)
- Management system controller with RJ-45 serial and 10/100 Mb/sec Ethernet ports

### Manufacture and install

- Internal disk: Up to four 73 GB or 146 GB 2.5 in., 10,000 rpm SAS disks, providing a maximum capacity of 584 GB; internal hardware RAID 1 (mirroring) and RAID 0 (striping) supported
- Internal DVD: One slimline DVD-ROM/CD-RW
- External storage: Sun offers a complete line of best-in-class, innovative storage hardware, software, and solutions—including tape drives, tape libraries, disk storage systems, data management software, and more. For more information, go to: [sun.com/storage/tek](http://sun.com/storage/tek).

### Environment

- Operating temperature:
  - Sea level to 900m (3,000 ft.): 5°C to 35°C
  - Above 900m (3,000 ft.): Decrease maximum temperature as altitude increases, 1°C/300m (1.6°F/1,000 ft.)
  - 5°C to 35°C
- Nonoperating temperature:
  - Sea level to 900m (3,000 ft.): 40°C to 80°C
  - Above 900m (3,000 ft.): 40°C to 60°C
- Operating relative humidity: 20-80%, noncondensing, 27°C wet bulb
- Nonoperating relative humidity: 98%, noncondensing
- Operating altitude: Up to 3,000m (10,000 ft.)
- Nonoperating altitude: Up to 12,000m (40,000 ft.)
- Acoustic noise:
  - Operating/idling: 7.7 B (low-dB 1 B = 10 dB)
  - Operating/idling: 62 dB (1 pAm, bystander positions)

### Power

- Dual-redundant, hot-swappable power supplies
- Typical operating power: 275W (note: earlier models containing 550W power supplies have a typical operating power of 335W)
- AC power: 100-240 V AC, 50-60 Hz (4.5 A at 100-120 V AC, 2.35 A at 200-240 V AC)

### Learn More

To learn more about the Sun Fire T2000 server, go to: [sun.com/T2000](http://sun.com/T2000).

### Regulations

Safety: UL/CSA 60950-1, EN 60950-1, IEC 60950-1 CB Scheme with all country deviations, IEC 825-1, 2, and CFR 21 part 1040, CNS 14336, GB 4943

Ergonomics: EN 1717-2000

RF/EMC: EN 55022 Class A, 47 CFR 15B Class A, ICES-003 Class A, VCCI Class A, AS/NZ 3548 Class A, CNS 13438 Class A, KSC 5838 Class A, GB 9254 Class A, EN 61000-3-2, EN 61000-3-3, GB 17625-1

Immunity: EN 55024/CISPR 24, IEC 6100-4-2, IEC 6100-4-3, IEC 6100-4-4, IEC 6100-4-5, IEC 6100-4-6, IEC 6100-4-8, IEC 6100-4-11

Telecommunications: EN 300 388

Regulatory markings: CE, FCC, ICES-003, C-Tick, VCCI, GOST R, NRC, UL-CUL, UL/DEMKO/GS, UL/5-Mark, BSMI, CCC

### Dimensions and weight

Height: 89 mm (3.5 in.) 2U

Weight: 425 mm (16.7 in.)

Depth: 621 mm (24.4 in.)

Weight: 18.3 kg (40 lbs.) approximate — includes two power supplies, two disk drives, one DVD

### Warranty

Hardware support: One year

Software install: 90 days

Call response: Eight hours

Delivery: Next business day

### Upgrade

Customers can trade in their old Sun or non-Sun servers and receive a discount toward the price of their new Sun Fire T2000 server. For more details on the Sun Upgrade Advantage Program, go to: [sun.com/ibh/coolthreads](http://sun.com/ibh/coolthreads).

<sup>1</sup> Actual capacity, weight, and performance is defined by the drive's performance capabilities.



Sun Microsystems, Inc. 4050 Network Circle, Santa Clara, CA 95054 USA Phone 1-800-950-5300 or 1-800-555-9513 Web [sun.com](http://sun.com)

© 2007 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, CoolThreads, Java, StorageTek, and Sun Fire are trademarks or registered trademarks of Sun Microsystems, Inc., or its subsidiaries in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc., in the United States and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the United States and other countries, which will be licensed through S.O. Open Company Ltd. Information subject to change without notice. SunWarranty 1.1.020002.01.04.07.08



Esta hoja de datos puede ser consultada en la siguiente liga (Enero 2011):

<http://www.sun.com/servers/coolthreads/t2000/datasheet.pdf>

Es un documento público y disponible en un formato con mayor calidad de imagen.

## ANEXO 2 - Instalación del Sistema Operativo Base

El instalador en el servidor SPARC por default inicia en modo texto, y es esta opción la que se utiliza, la primera pantalla de configuración del sistema y requiere que se elija el idioma del sistema y de la interfaz de instalación, se selecciona la opción 0 correspondiente a **Inglés**.

```

Select a Language
0. English
1. French
2. German
3. Italian
4. Japanese
5. Korean
6. Simplified Chinese
7. Spanish
8. Swedish
9. Traditional Chinese

Please make a choice (0 - 9), or press h or ? for help: 0

```

Se elige el tipo de terminal que se está utilizando, la cual es la **xterms**, por ello se selecciona la opción **12**.

```

What type of terminal are you using?
1) ANSI Standard CRT
2) DEC VT52
3) DEC VT100
4) Heathkit 19
5) Lear Siegler ADM31
6) PC Console
7) Sun Command Tool
8) Sun Workstation
9) Televideo 910
10) Televideo 925
11) Wyse Model 50
12) X Terminal Emulator (xterms)
13) CDE Terminal Emulator (ditterm)
14) Other
Type the number of your choice and press Return: 12
Completing system identification...
in rdisc: No interfaces up

```

La siguiente pantalla informa que si se trata de un sistema cableado, como es en este caso, de forma automática va a recabar información sin embargo se le deben especificar ciertos parámetros de red, para que se comience con el análisis se presiona **F2**.

```

F2_Continue   F6_Help
Identify This System _____

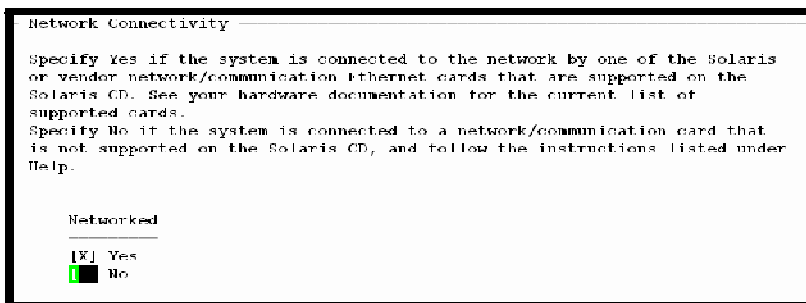
On the next screens, you must identify this system as networked or
non-networked, and set the default time zone and date/time.

If this system is networked, the software will try to find the information
it needs to identify your system; you will be prompted to supply any
information it cannot find.

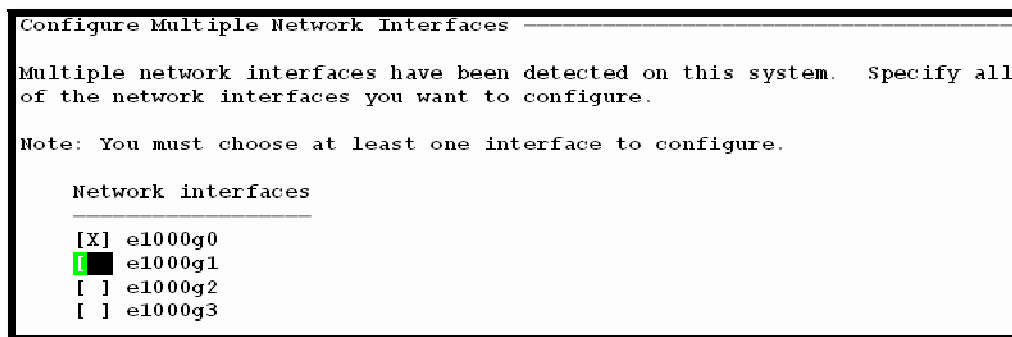
> To begin identifying this system, press F2.

```

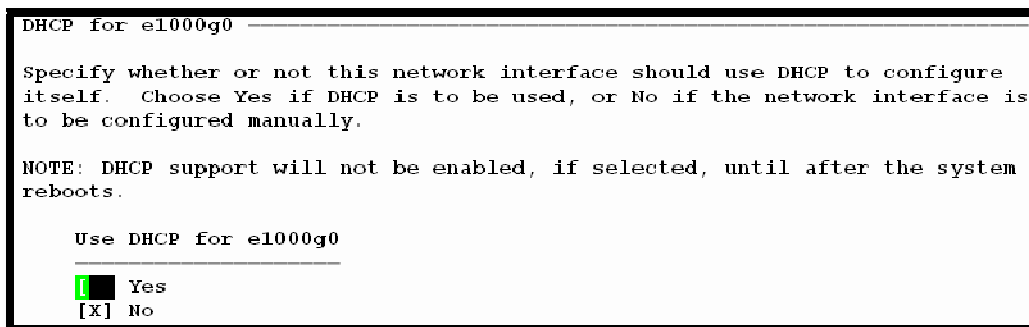
En la sección Network Connectivity es necesario indicar si el equipo está conectado a la red, tomando en cuenta que dicha interfaz del servidor es soportada por Solaris se selecciona la casilla **Yes**.



El instalador identifica las interfaces de red con las que cuenta el equipo, en esta sección se debe indicar la interfaz o interfaces que se van a configurar. La interfaz a configurar es la **e1000g0**.



Los parámetros de red se pueden asignar de 2 formas, de forma automática donde un servidor DHCP se encarga de asignar la Ip, máscara de subred de manera automática o manual donde todos estos parámetros deben especificarse en pantallas posteriores. Le indicamos que se realizarán de forma manual mediante **No**.



En las pantallas siguientes proporcionamos el nombre del servidor el cual debe ser único, en este caso se le asigna el nombre **UXSOP001**.

```
Host Name for e1000g0 -----
Enter the host name which identifies this system on the network. The name
must be unique within your domain; creating a duplicate host name will cause
problems on the network after you install Solaris.

A host name must have at least one character; it can contain letters,
digits, and minus signs (-).

Host name for e1000g0 uxsop001
```

En esta pantalla se establece la Ip que va a utilizar la interfaz e1000g0, la cual es **192.9.198.190**.

```
IP Address for e1000g0 -----
Enter the Internet Protocol (IP) address for this network interface. It
must be unique and follow your site's address conventions, or a
system/network failure could result.

IP addresses contain four sets of numbers separated by periods (for example
129.200.9.1).

IP address for e1000g0 192.9.198.190
```

El equipo forma parte de una subred y es necesario establecerlo mediante la selección de **Yes**.

```
Subnet for e1000g0 -----
On this screen you must specify whether this system is part of a subnet. If
you specify incorrectly, the system will have problems communicating on the
network after you reboot.

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

System part of a subnet
-----
[X] Yes
[ ] No
```

Se le indica la máscara de subred, ya que la ip es de la clase C entonces le corresponde la máscara **255.255.255.0**.



```

Netmask for e1000g0 -----

On this screen you must specify the netmask of your subnet.  A default
netmask is shown; do not accept the default unless you are sure it is
correct for your subnet.  A netmask must contain four sets of numbers
separated by periods (for example 255.255.255.0).

Netmask for e1000g0 255.255.255.0

```

La conexión que se utiliza en internet convencional es por ello que no se va utilizar el protocolo IPv6 el cual se utiliza en internet 2, se le indica con **No**.

```

IPv6 for e1000g0 -----

Specify whether or not you want to enable IPv6, the next generation Internet
Protocol, on this network interface.  Enabling IPv6 will have no effect if
this machine is not on a network that provides IPv6 service.  IPv4 service
will not be affected if IPv6 is enabled.

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Enable IPv6 for e1000g0
-----
[ ] Yes
[X] No

```

Posteriormente se especifica que se cuenta con un Ruteador, el cual se encarga de todo el ruteo mediante **Specify One**.

```

Set the Default Route for e1000g0 -----

To specify the default route, you can let the software try to detect one
upon reboot, you can specify the IP address of the router, or you can choose
None.  Choose None if you do not have a router on your subnet.

> To make a selection, use the arrow keys to select your choice and press
Return to mark it [X].

Default Route for e1000g0
-----
[ ] Detect one upon reboot
[X] Specify one
[ ] None

```

A continuación se indica la dirección IP del Router que se va a utilizar, **192.9.198.1**.

```

Default Route IP Address for e1000g0 -----

Enter the IP address of the default route. This entry will be placed in the
/etc/defaultrouter file and will be the default route after you reboot
(example 129.146.89.225).

Router IP Address for e1000g0 192.9.198.1

```

Se muestra un resumen de la información que hasta el momento se ha proporcionado, una vez que se verifica que toda la información es correcta, proseguimos con la instalación presionando **F2**.

```
Confirm Information for e1000g0
> Confirm the following information.  If it is correct, press F2;
  to change any information, press F4.

      Networked: Yes
      Use DHCP: No
      Host name: uxsop001
      IP address: 192.9.198.190
system part of a subnet: Yes
      Netmask: 255.255.255.0
      Enable IPv6: No
      Default Route: Specify one
      Router IP Address: 192.9.198.1
```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No**.

```
Configure Security Policy:
Specify Yes if the system will use the Kerberos security mechanism.
Specify No if this system will use standard UNIX security.

Configure Kerberos Security
-----
[ ] Yes
[X] No
```

En la siguiente pantalla se solicita la confirmación que no va utilizar la seguridad mediante Kerberos para ello se teclea **F2**.

```
Confirm Information
> Confirm the following information.  If it is correct, press F2;
  to change any information, press F4.

Configure Kerberos Security: No
```

Si el servidor tendrá o no un servicio denominado Name Service, ya sea NIS, DNS o LDAP se selecciona en esta sección, se selecciona **None** ya que el servidor no tendrá ninguno de los servicios mencionados.

```
Name Service -----
On this screen you must provide name service information.  Select the name
service that will be used by this system, or None if your system will either
not use a name service at all, or if it will use a name service not listed
here.

> To make a selection, use the arrow keys to highlight the option
and press Return to mark it [X].

Name service
-----
[ ] NIS+
[ ] NIS
[ ] DNS
[ ] LDAP
[X] None
```

En la siguiente pantalla se confirma que no se va tener ningún servicio, esto presionando la tecla **F2**.

```
Confirm Information -----
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.

Name service: None
```

En la sección **NFSv4 Domain Name**, seleccionamos **Use the NFSv4 domain derived by the system** con lo cual se genera un nombre de dominio de forma automática el cual se utiliza para la resolución interna.

```
NFSv4 Domain Name -----
NFS version 4 uses a domain name that is automatically derived from the
system's naming services.  The derived domain name is sufficient for most
configurations.  In a few cases, mounts that cross domain boundaries might
cause files to appear to be owned by "nobody" due to the lack of a common
domain name.

The current NFSv4 default domain is: ""

NFSv4 Domain Configuration
-----
[X] Use the NFSv4 domain derived by the system
[ ] Specify a different NFSv4 domain
```

En la siguiente pantalla se confirma que va a utilizar **Use the NFSv4 domain derived by the system**, presionando la tecla **F2**.

```

Confirm Information for NFSv4 Domain -----
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.

NFSv4 Domain Name:  << Value to be derived dynamically >>

```

Se debe seleccionar la zona horaria y para ello nos da diferentes formas de elegirla, la opción que se elige es la de "Continente/País/Región Geográfica".

Se elige **Americas**, para especificar que el equipo se encuentra en América.

```

Time Zone -----
On this screen you must specify your default time zone.  You can specify a
time zone in three ways:  select one of the continents or oceans from the
list, select other - offset from GMT, or other - specify time zone file.

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Continent and Oceans
-----
-  [ ] Africa
  [X] Americas
  [ ] Antarctica
  [ ] Arctic Ocean
  [ ] Asia
  [ ] Atlantic Ocean
  [ ] Australia
  [ ] Europe
v  [ ] Indian Ocean

```

En la sección de Country or Region, se selecciona **Mexico**.

```

Country or Region -----
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Countries and Regions
-----
^  [ ] El Salvador
  [ ] French Guiana
  [ ] Greenland
  [ ] Grenada
  [ ] Guadeloupe
  [ ] Guatemala
  [ ] Guyana
  [ ] Haiti
  [ ] Honduras
  [ ] Jamaica
  [ ] Martinique
  [X] Mexico
v  [ ] Montserrat

Esc-2_Continue  Esc-6_Help

```

En Time Zone se indica **Central Time** para la ciudad de México.

```

Time Zone -----
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Time zones
-----
[ X] Central Time - most locations
[ ] Central Time - Quintana Roo
[ ] Central Time - Campeche, Yucatan
[ ] Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas
[ ] Mountain Time - S Baja, Nayarit, Sinaloa
[ ] Mountain Time - Chihuahua
[ ] Mountain Standard Time - Sonora
[ ] Pacific Time

```

En la siguiente pantalla se le proporciona la fecha y hora y posteriormente se confirma esta información mediante **F2**.

```

Date and Time -----
> Accept the default date and time or enter
new values.

Date and time: 2010-09-10 15:37

Year (4 digits) : 2010
Month (1-12) : 09
Day (1-31) : 10
Hour (0-23) : 15
Minute (0-59) : 37

```

```

Confirm Information -----
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Time zone: Central Time - most locations
(Mexico/General)
Date and time: 2010-09-10 15:37:00

```

El usuario root es el que tiene el mayor nivel de privilegios y es por esto que el instalador solicita el **password de root**, el password debe definir basándose en las buenas prácticas para la selección de contraseñas de tal forma que no pueda ser fácilmente descifrada.

```

Root Password -----
Please enter the root password for this system.

The root password may contain alphanumeric and special characters. For
security, the password will not be displayed on the screen as you type it.

> If you do not want a root password, leave both entries blank.

Root password: *****
Root password: *****

```

La configuración para **Remote Services Enabled**, que son todos los servicios de red para clientes remotos, por razones de seguridad seleccionamos la opción de **NO**, donde deshabilitamos estos servicios con la excepción de Secure Socket Layer (SSH), esto se realiza con la finalidad de reducir la superficie de ataque mediante la desactivación de los servicios de red que no se van a utilizar y que de ser necesarios se configuran de forma independiente después de la finalización de la instalación

```
Enabling remote services

Would you like to enable network services for use by remote clients?

Selecting "No" provides a more secure configuration in
which Secure Shell is the only network service provided to
remote clients. Selecting "Yes" enables a larger set of
services as in previous Solaris releases. If in doubt, it is
safe to select "No" as any services can be individually enabled
after installation.

Note: This choice only affects initial installs. It doesn't affect upgrades.

Remote services enabled
-----
| | Yes
| F | No
```

Se muestra el progreso de la instalación **System identificación Complete**, así como los procesos que se encuentra ejecutando.

```
System identification complete.
Starting Solaris installation program...
Executing JumpStart preinstall phase...
Searching for SolStart directory...
Checking rules.ok file...
Using begin script: install_begin
Using finish script: patch_finish
Executing SolStart preinstall phase...
Executing begin script "install_begin"...
Begin script install_begin execution completed.
```

La instalación continúa seleccionando el tipo de instalación que se ofrece, **Standard** incluida en Solaris para un sistema inicial o **Flash Archive** la cual consiste en instalar y configurar un respaldo previamente hecho de un sistema base. Se selecciona Standard mediante la tecla **F2**.

```

Solaris Interactive Installation
-----
On the following screens, you can accept the defaults or you can customize
how Solaris software will be installed by:

- Selecting the type of Solaris software to install
- Selecting disks to hold software you've selected
- Selecting unbundled products to be installed with Solaris
- Specifying how file systems are laid out on the disks

After completing these tasks, a summary of your selections (called a
profile) will be displayed.

There are two ways to install your Solaris software:

- "Standard" installs your system from a standard Solaris Distribution.
  Selecting "Standard" allows you to choose between initial install
  and upgrade, if your system is upgradable.
- "Flash" installs your system from one or more Flash Archives.

-----
F2_Standard   F4_Flash   F5_Exit   F6_Help

```

Con la finalidad que se expulse de forma automática el CD de instalación, se selecciona **Automatically eject CD/DVD** y se tecléa **F2** para continuar.

```

Eject a CD/DVD Automatically?
-----
During the installation of Solaris software, you may be using one or more
CDs/DVDs. You can choose to have the system eject each CD/DVD automatically
after it is installed or you can choose to manually eject each CD/DVD.

 Automatically eject CD/DVD
 Manually eject CD/DVD

```

Debido a que no se van instalar nada adicional al equipo, se selecciona **Auto Reboot** y se tecléa **F2** para continuar.

```

Reboot After Installation?
-----
After Solaris software is installed, the system must be rebooted. You can
choose to have the system automatically reboot, or you can choose to
manually reboot the system if you want to run scripts or do other
customizations before the reboot. You can manually reboot a system by using
the reboot(1M) command.

 Auto Reboot
 Manual Reboot

-----
F2_Continue   F3_Go Back   F5_Exit

```

El sistema inicializa para continuar con la configuración, esto puede tardar varios minutos, al igual que comienza a descargar el medio de instalación.

```

Initializing
-----

The system is being initialized.
  
```

Se muestra la licencia, posterior a revisarla y si se está de acuerdo, se tecléa **F2** para aceptarla y continuar con la instalación.

```

License
-----

Sun Microsystems, Inc. ("Sun")
SOFTWARE LICENSE AGREEMENT

READ THE TERMS OF THIS AGREEMENT ("AGREEMENT") CAREFULLY BEFORE
OPENING SOFTWARE MEDIA PACKAGE. BY OPENING SOFTWARE MEDIA
PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCESSING SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF
THESE TERMS BY SELECTING THE "ACCEPT" (OR EQUIVALENT) BUTTON AT
THE END OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE
TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF
PURCHASE FOR A REFUND OR, IF SOFTWARE IS ACCESSED ELECTRONICALLY,
SELECT THE "DECLINE" (OR EQUIVALENT) BUTTON AT THE END OF THIS
AGREEMENT. IF YOU HAVE SEPARATELY AGREED TO LICENSE TERMS
("MASTER TERMS") FOR YOUR LICENSE TO THIS SOFTWARE, THEN SECTIONS
1-6 OF THIS AGREEMENT ("SUPPLEMENTAL LICENSE TERMS") SHALL
SUPPLEMENT AND SUPERSEDE THE MASTER TERMS IN RELATION TO THIS
SOFTWARE.

v
1. Definitions.

F2_Accept License    F5_Exit
  
```

En la región geográfica se selecciona **Mexico (ISO8859-1)**.

```

Select Geographic Regions
-----

Select the geographic regions for which support should be installed.

> [ ] Northern Europe
> [ ] Northern Africa
> [ ] Middle East
> [ ] Southern Europe
> [ ] South America
> [ ] Central America
> [ ] Central Europe
v [ / ] North America
  [ ] Canada-English (ISO8859-1)
  [ ] Canada-French (ISO8859-1)
  [ ] French
  [x] Mexico (ISO8859-1)
  [ ] Spanish
  [ ] U.S.A. (UTF-8)
  [ ] U.S.A. (en_US.ISO8859-1)
> [ ] Western Europe

Locale is selected. Press Return to deselect

F2_Continue    F3_Go Back    F5_Exit    F6_Help
  
```



Se selecciona **North America** en Locale, locale contiene una colección de archivos, datos y código, información necesaria para adaptar Solaris a un mercado geográfico específico como son los mensajes mostrados al usuario, información de los componentes de la interfaz, fechas, formatos de moneda, y las fuentes u otra información específica por escrito entre otros.

```
Select System Locale
-----
Select the initial locale to be used after the system has been installed.

[X] POSIX C ( C )
    North America
[ ] Mexico (ISO8859-1) ( es_MX.ISO8859-1 )
[ ] Spanish, Mexico (UTF-8) ( es_MX.UTF-8 )
```

Debido a que no se va instalar ningún producto adicional, se debe indicar con **None**.

```
Additional Products
-----
To scan for additional products, select the location you wish to scan.
Products found at the selected location that are in a Web Start Ready
install form will be added to the Products list.

Web Start Ready product scan location:

[X] None
[ ] CD/DVD
[ ] Network File System
```

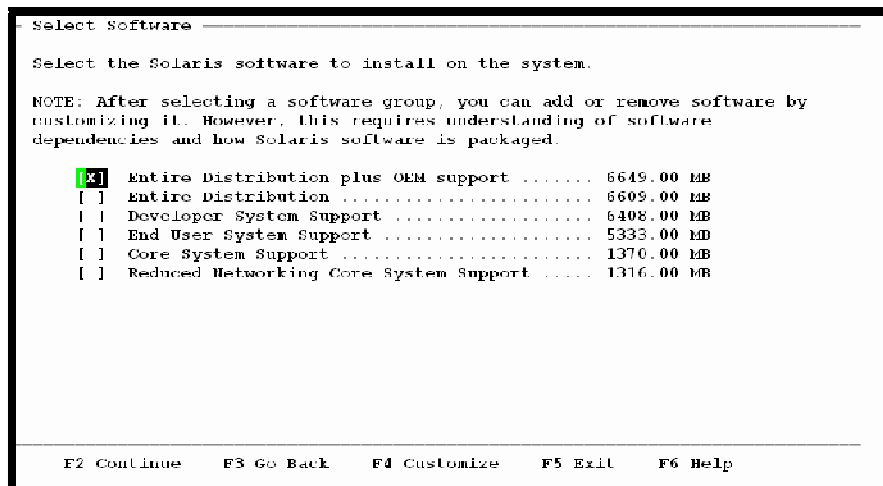
Se procede a seleccionar el *Filesystem* que se va a utilizar, se selecciona **UFS** que es el nativo en UNIX para instalación del Sistema Operativo y o **ZFS** ya que este último se utilizará en las zonas ya que ofrece un mayor espacio para los archivos, mejora enormemente la administración y mejorado en gran medida la seguridad de datos.

```
Choose Filesystem Type
-----
Select the filesystem to use for your Solaris installation

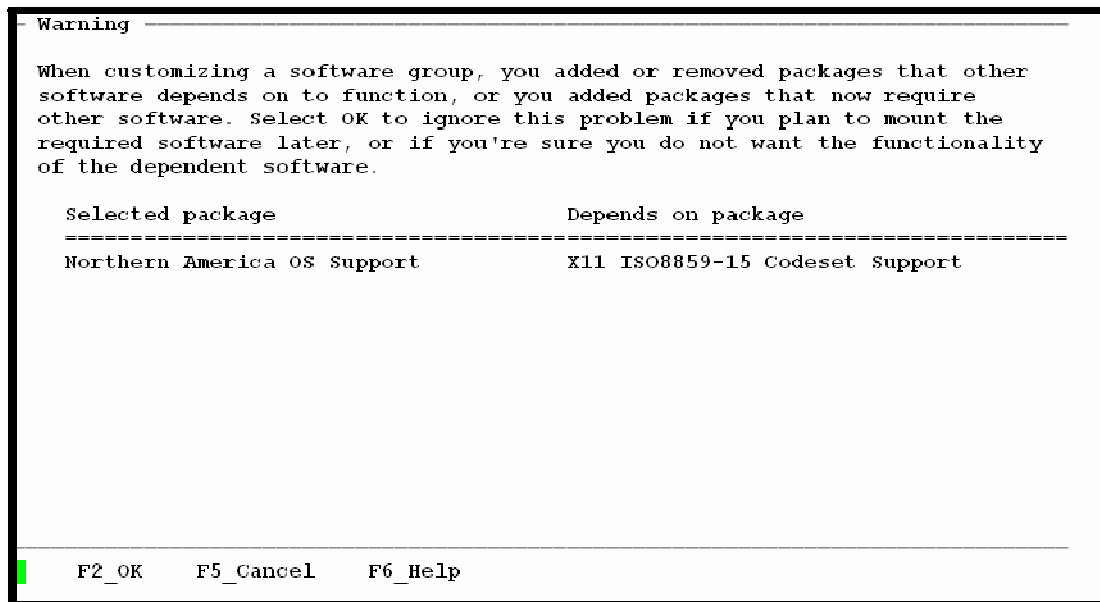
[X] UFS
[ ] ZFS

F2_Continue  F3_Go Back  F5_Exit  F6_Help
```

En la selección del sistema de Solaris, se prefiere la opción de **Entire Distribution plus OEM support** del menú de software, ya que esta opción incluye controladores de hardware adicionales incluso para hardware que se va incorporar posteriormente a la instalación, además de los paquetes para el grupo de software de distribución completa.



Aparece una pantalla de advertencia (Warning) indica que se seleccionó **Locale Northern America OS Support** y de su dependencia con ese paquete de información. Se tecllea F2 para continuar con el proceso.



El disco duro del servidor donde se llevará a cabo la instalación es **clt0d0**, se selecciona este disco y se tecllea **F2** para continuar con el proceso.

```

Select Disks -----

On this screen you must select the disks for installing Solaris software.
Start by looking at the Suggested Minimum field; this value is the
approximate space needed to install the software you've selected. Keep
selecting disks until the Total Selected value exceeds the Suggested Minimum
value.
NOTE: ** denotes current boot disk

Disk Device                                     Available Space
-----
[X] ** c1t0d0                                     139989 MB (F4 to edit

                                     Total Selected: 139989 MB
                                     Suggested Minimum:  5217 MB

-----
F2_Continue  F3_Go Back  F4_Edit  F5_Exit  F6_Help

```

Posteriormente se selecciona el disco donde se instalará el directorio raíz, el cual es **c1t0d0** y se teclea **F2** para continuar.

```

Select Boot Disk -----

On this screen you can select the disk for installing the root (/) file
system of the Solaris software.

Original Boot Device : c1t0d0

      Disk
      =====
[X] c1t0d0 (F4 to select boot device)

```

Adicionalmente configuramos los sistemas de archivos para ellos se ofrecen las opciones de **Auto Layout** esta opción a hacer una sola partición de disco de arranque completo mientras que en **Manual Layout** se puede asignar los recursos necesarios para cada partición, preferimos la segunda opción ya que es más granular y con esto se distribuyen mejor los recursos lo que da lugar a un mejor mayor eficiencia y se indica con la tecla **F4**.

```
Automatically Layout File Systems? -----
Do you want to use auto-layout to automatically layout file systems?
Manually laying out file systems requires advanced system administration
skills.

F2_Auto Layout   F3_Go Back   F4_Manual Layout   F5_Exit   F6_Help
```

En primera instancia se indica la distribución actual y se tecléa F2 para iniciar la configuración.

```
File System and Disk Layout -----
The summary below is your current file system and disk layout, based on the
information you've supplied.

NOTE: If you choose to customize, you should understand file systems, their
intended purpose on the disk, and how changing them may affect the operation
of the system.

File sys/mnt point Disk/Slice                               Size
-----
overlap          c1t0d0s2                               139989 MB

F2_Continue   F3_Go Back   F4_Customize   F5_Exit   F6_Help
```

Haciendo uso del cursor se mueve en el número de partición que se desea configurar, posteriormente proporciona la información del directorio y su respectivo tamaño, quedando la distribución de la siguiente forma:

```

File System and Disk Layout
-----
The summary below is your current file system and disk layout, based on the
information you've supplied.

NOTE: If you choose to customize, you should understand file systems, their
intended purpose on the disk, and how changing them may affect the operation
of the system.

File sys/Mnt point Disk/Slice                               Size
-----
/                   c1t0d0s0                               61453 MB
swap                c1t0d0s1                               16386 MB
overlap             c1t0d0s2                               139989 MB
                   c1t0d0s3                               8198 MB
/var/crash          c1t0d0s4                               8198 MB
                   c1t0d0s6                               45493 MB
                   c1t0d0s7                               258 MB

F2_Continue   F3_Go Back   F4_Customize   F5_Exit   F6_Help

```

La instalación no requiere la ayuda de un servidor adicional por lo que se prosigue con la instalación mediante **F2**.

```

Mount Remote File Systems?
-----
Do you want to mount software from a remote file server? This may be
necessary if you had to remove software because of disk space problems.

```

En la pantalla de Profile se indica un resumen de las configuración, ya que esta información es correcta se tecléa **F2** para continuar.

```

Profile
-----
The information shown below is your profile for installing Solaris software.
It reflects the choices you've made on previous screens.

-----

Installation Option: Initial
Boot Device: c1t0d0
Root File System Type: UFS
Client Services: None

Locales: Mexico (ISO8859-1)
System Locale: C ( C )

Software: Solaris 10, Entire Distribution plus OEM su

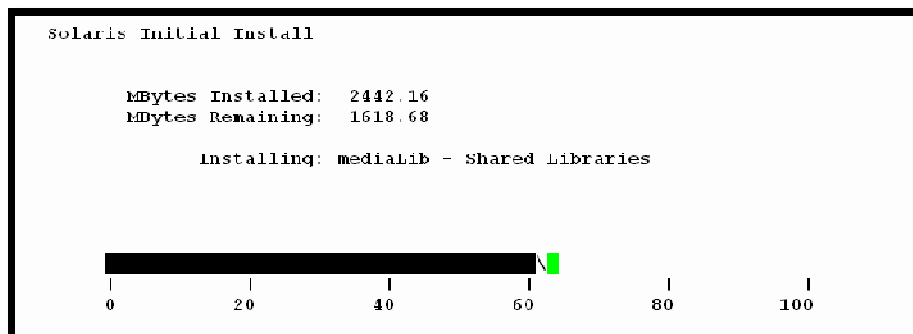
File System and Disk Layout: /                   c1t0d0s0 61453 MB
                             swap                c1t0d0s1 16386 MB
                             /var/crash          c1t0d0s4 8198 MB

F2_Begin Installation   F4_Change   F5_Exit   F6_Help

```

La instalación comienza, se informa de los cambios que se realizaron en disco duro posteriormente el instalador muestra el número de bytes instalados al igual que los

faltantes, así como una breve descripción de la herramienta que se encuentra instalando y en la parte inferior el porcentaje de instalación en unas escala de 0 a 100, donde 100 % es la finalización del proceso de instalación.



El servidor se reinicia de forma automática y comienza con la configuración de los dispositivos.

```
continue. [c]
Creating boot_archive for /a
updating /a/platform/sun4v/boot_archive
15+0 records in
15+0 records out
syncing file systems... done
rebooting...

SC Alert: Host System has Reset
/

Sun Fire T200, No Keyboard
Copyright 2010 Sun Microsystems, Inc. All rights reserved.
OpenBoot 4.30.4 a, 8064 MB memory available, Serial #64551886.
Ethernet address 0:3:ba:d8:1b:ce, Host ID: 83d81bce.

Boot device: disk File and args:
SunOS Release 5.10 Version Generic_141444 09 64 bit
Copyright 1983 2009 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
Hostname: uxsop001
Configuring devices.
```

Finalmente se indica el idioma del Teclado, el cual es US – Inglés.

```
Configure Keyboard Layout
-----
Please specify the keyboard layout from the list below.

> To make a selection, use the arrow keys to highlight the option and
  press Return to mark it [X].

Keyboard Layout
-----
^
| [ ] Slovenian
| [ ] Slovakian
| [ ] Spanish
| [ ] Swedish
| [ ] Swiss-French
| [ ] Swiss-German
| [ ] Taiwanese
| [ ] TurkishQ
| [ ] TurkishF
| [ ] UK-English
- [X] US-English
-----
F2_Continue    F6_Help
```

## ANEXO 3 - Configuración de los Solaris Containers

### Zona Ecommerce -Db

Se establece el nombre del host, para el cual se especifica ecommerce -db y se teclea F2 para confirmar y continuar.

```

Enter the host name which identifies this system on the network. The name
must be unique within your domain; creating a duplicate host name will cause
problems on the network after you install Solaris.

A host name must have at least one character; it can contain letters,
digits, and minus signs (-).

Host name for e1000g0:4 ecommerce-db

```

Se confirma el nombre del Host con la tecla F2.

```

> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Host name: ecommerce-db

```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No** y posteriormente confirmando con la tecla **F2**.

```

Specify Yes if the system will use the Kerberos security mechanism.

Specify No if this system will use standard UNIX security.

Configure Kerberos Security
#####
[ ] Yes
[X] No

```







En Time Zone se indica **Central Time** para la ciudad de México, posteriormente se confirma esta zona con la tecla **F2**.

```
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Time zones
#####
[X] Central Time - most locations
[ ] Central Time - Quintana Roo
[ ] Central Time - Campeche, Yucatan
[ ] Mexican Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas awa
[ ] US Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas near US
[ ] Mountain Time - S Baja, Nayarit, Sinaloa
[ ] Mexican Mountain Time - Chihuahua away from US border
[ ] US Mountain Time - Chihuahua near US border
[ ] Mountain Standard Time - Sonora
[ ] US Pacific Time - Baja California near US border
[ ] Mexican Pacific Time - Baja California away from US border
[ ] Mexican Central Time - Bahia de Banderas
```

```
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Time zone: Central Time - most locations
(Mexico/General)
```

El usuario root es el que tiene el mayor nivel de privilegios y es por esto que el instalador solicita el **password de root**, el password debe definirse basándose en las buenas prácticas para la selección de contraseñas de tal forma que no pueda ser fácilmente descifrada, posteriormente para continuar tecleamos **Esc** y **2**.

```
Please enter the root password for this system.

The root password may contain alphanumeric and special characters. For
security, the password will not be displayed on the screen as you type it.

> If you do not want a root password, leave both entries blank.

Root password: *****
Root password: *****
```

Se nos indica que la zona va a reiniciar debido a la configuración previamente hecha, estos cambios se realizaron en **/etc/default/init**.

```
rebooting system due to change(s) in /etc/default/init
```

```
[NOTICE: Zone rebooting]
```

```
SunOS Release 5.10 Version Generic_142909-17 64-bit  
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.  
Hostname: ecommerce-db  
Reading ZFS config: done.
```

```
[NOTICE: Zone rebooting]
```

```
SunOS Release 5.10 Version Generic_142909-17 64-bit  
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.  
Hostname: ecommerce-db  
Reading ZFS config: done.
```

```
ecommerce-db console login: Oct 31 14:48:30 ecommerce-db sendmail[11468]: My unq  
ualified host name (ecommerce-db) unknown; sleeping for retry  
Oct 31 14:48:30 ecommerce-db sendmail[11470]: My unqualified host name (ecommerc  
e-db) unknown; sleeping for retry  
Oct 31 14:49:30 ecommerce-db sendmail[11468]: unable to qualify my own domain na  
me (ecommerce-db) -- using short name  
Oct 31 14:49:30 ecommerce-db sendmail[11470]: unable to qualify my own domain na  
me (ecommerce-db) -- using short name
```

```
ecommerce-db console login: root  
Password:  
Oct 31 14:54:44 ecommerce-db login: ROOT LOGIN /dev/console  
Last login: Sat Oct 30 14:58:19 on pts/6  
Sun Microsystems Inc. SunOS 5.10 Generic January 2005  
# █
```

Finalmente que los parámetros de nombre de host, ip y el nombre sean los correctos, se verifica en **/etc/hosts**



## Zona Ecommerce -App

Se establece el nombre del host, para el cual se especifica ecommerce –app y se tecldea **F2** para continuar.

```
Enter the host name which identifies this system on the network. The name
must be unique within your domain; creating a duplicate host name will cause
problems on the network after you install Solaris.

A host name must have at least one character; it can contain letters,
digits, and minus signs (-).

Host name for e1000g0:5 ecommerce-app
```

Se confirma el nombre del Host con la tecla **F2**.

```
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Host name: ecommerce-app
```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No** y posteriormente confirmando con la tecla **F2**.

```
Specify Yes if the system will use the Kerberos security mechanism.

Specify No if this system will use standard UNIX security.

Configure Kerberos Security
#####
[ ] Yes
[X] No
```

```
Specify Yes if the system will use the Kerberos security mechanism.

Specify No if this system will use standard UNIX security.

Configure Kerberos Security
#####
[ ] Yes
[X] No
```

Si el servidor tendrá o no un servicio denominado Name Service, ya sea NIS, DNS o LDAP se selecciona en esta sección, se selecciona **None** ya que el servidor no tendrá



```
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.
```

```
NFSv4 Domain Name:  << Value to be derived dynamically >>
```

Se debe seleccionar la zona horaria y para ello nos da diferentes formas de elegirla, la opción que se elige es la de “Continente/País/Región Geográfica”.

Se elige **Americas**, para especificar que el equipo se encuentra en América.

```
On this screen you must specify your default time zone.  You can specify a
time zone in three ways:  select one of the continents or oceans from the
list, select other - offset from GMT, or other - specify time zone file.
```

```
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].
```

```
Continents and Oceans
#####
-  [ ] Africa
x  [X] Americas
x  [ ] Antarctica
x  [ ] Arctic Ocean
x  [ ] Asia
x  [ ] Atlantic Ocean
x  [ ] Australia
x  [ ] Europe
v  [ ] Indian Ocean
```

En la sección de Country or Region, se selecciona **Mexico**.

```
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].
```

```
Countries and Regions
#####
^  [ ] El Salvador
x  [ ] French Guiana
x  [ ] Greenland
x  [ ] Grenada
x  [ ] Guadeloupe
x  [ ] Guatemala
x  [ ] Guyana
x  [ ] Haiti
x  [ ] Honduras
x  [ ] Jamaica
x  [ ] Martinique
x  [X] Mexico
v  [ ] Montserrat
```

En Time Zone se indica **Central Time** para la ciudad de México, posteriormente se confirma esta zona con la tecla **F2**.





```
rebooting system due to change(s) in /etc/default/init

[NOTICE: Zone rebooting]

SunOS Release 5.10 Version Generic_142909-17 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.
Hostname: ecommerce-app
Reading ZFS config: done.

ecommerce-app console login: █
```

Finalmente que los parámetros de nombre de host, ip y el nombre sean los correctos, se verifica en **/etc/hosts**

```
bash-3.00# hostname
ecommerce-app
bash-3.00# ifconfig -a
lo0:4: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
e1000g0:5: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.9.198.235 netmask ffffffff broadcast 192.9.198.255
bash-3.00# █
```

## Zona Colaboracion -App

Se establece el nombre del host, para el cual se especifica colaboracion –app y se tecldea **F2** para continuar.

```
Enter the host name which identifies this system on the network. The name
must be unique within your domain; creating a duplicate host name will cause
problems on the network after you install Solaris.

A host name must have at least one character; it can contain letters,
digits, and minus signs (-).

Host name for e1000g0:3 colaboracion-app
```

Se confirma el nombre del Host con la tecla **F2**.

```
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Host name: colaboracion-app
```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No** y posteriormente confirmando con la tecla **F2**.

```
Specify Yes if the system will use the Kerberos security mechanism.
Specify No if this system will use standard UNIX security.

Configure Kerberos Security
#####
[ ] Yes
[X] No
```

```
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Configure Kerberos Security: No
```

Si el servidor tendrá o no un servicio denominado Name Service, ya sea NIS, DNS o LDAP se selecciona en esta sección, se selecciona **None** ya que el servidor no tendrá

ninguno de los servicios mencionados y posteriormente se confirma esta opción confirmando con la tecla **F2**.

```
On this screen you must provide name service information.  Select the name
service that will be used by this system, or None if your system will either
not use a name service at all, or if it will use a name service not listed
here.
```

```
> To make a selection, use the arrow keys to highlight the option
and press Return to mark it [X].
```

```
Name service
qqqqqqqqqqqqqq
[ ] NIS+
[ ] NIS
[ ] DNS
[ ] LDAP
[X] None
```

```
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.
```

```
Name service: None
```

En la sección **NFSv4 Domain Name**, seleccionamos **Use the NFSv4 domain derived by the system** con lo cual se genera un nombre de dominio de forma automática el cual se utiliza para la resolución interna y posteriormente se confirma esta opción con la tecla **F2**.

```
NFS version 4 uses a domain name that is automatically derived from the
system's naming services.  The derived domain name is sufficient for most
configurations.  In a few cases, mounts that cross domain boundaries might
cause files to appear to be owned by "nobody" due to the lack of a common
domain name.
```

```
The current NFSv4 default domain is: ""
```

```
NFSv4 Domain Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
[X] Use the NFSv4 domain derived by the system
[ ] Specify a different NFSv4 domain
```

```
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.
```

```
NFSv4 Domain Name: << Value to be derived dynamically >>
```





```
rebooting system due to change(s) in /etc/default/init

[NOTICE: Zone rebooting]

SunOS Release 5.10 Version Generic_142909-17 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.
Hostname: colaboracion-app
Reading ZFS config: done.

colaboracion-app console login: █

Connected to 192.9.198.190                               SSH2 - ses128-cbc - hmac-sha2
```

Finalmente que los parámetros de nombre de host, ip y el nombre sean los correctos, se verifica en **/etc/hosts**

```
bash-3.00# hostname
colaboracion-app
bash-3.00# ifconfig -a
lo0:2: flags=2001000849<UP, LOOPBACK, RUNNING, MULTICAST, IPv4, VIRTUAL> mtu 8232 index 1
    inet 127.0.0.1 netmask ff000000
e1000j0:3: flags=1000043<UP, BROADCAST, RUNNING, MULTICAST, IPv4> mtu 1500 index 2
    inet 192.9.198.185 netmask ffffffff0 broadcast 192.9.198.255
bash-3.00# █
```

### Zona Colaboracion -Db

Se establece el nombre del host, para el cual se especifica colaboracion-db y se tecllea F2 para continuar.

```

for e1000g0:2
----- Description -----
Enter the host name which identifies this system on the network. The name
must be unique within your domain; creating a duplicate host name will cause
problems on the network after you install Solaris.

A host name must have at least one character; it can contain letters,
digits, and minus signs (-).
duplicate host names can cause network problems, do not guess;

Host name for e1000g0:2 colaboracion-db

```

Se confirma el nombre del Host con la tecla F2.

```

to change any information, press F4.

Host name: colaboracion-db

```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No** y posteriormente confirmando con la tecla F2.

```

Specify Yes if the system will use the Kerberos security mechanism.
Specify No if this system will use standard UNIX security.

Configure Kerberos Security
#####
[ ] Yes
[X] No

```

```

> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Configure Kerberos Security: No

```

Si el servidor tendrá o no un servicio denominado Name Service, ya sea NIS, DNS o LDAP se selecciona en esta sección, se selecciona **None** ya que el servidor no tendrá



ninguno de los servicios mencionados y posteriormente se confirma esta opción confirmando con la tecla **F2**.

```
On this screen you must provide name service information.  Select the name
service that will be used by this system, or None if your system will either
not use a name service at all, or if it will use a name service not listed
here.

> To make a selection, use the arrow keys to highlight the option
and press Return to mark it [X].

Name service
#####
[ ] NIS+
[ ] NIS
[ ] DNS
[ ] LDAP
[X] None
```

```
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.
```

```
Name service: None
```

En la sección **NFSv4 Domain Name**, seleccionamos **Use the NFSv4 domain derived by the system** con lo cual se genera un nombre de dominio de forma automática el cual se utiliza para la resolución interna y posteriormente se confirma esta opción con la tecla **F2**.

```
NFS version 4 uses a domain name that is automatically derived from the
system's naming services. The derived domain name is sufficient for most
configurations. In a few cases, mounts that cross domain boundaries might
cause files to appear to be owned by "nobody" due to the lack of a common
domain name.
```

```
The current NFSv4 default domain is: ""
```

```
NFSv4 Domain Configuration
#####
[X] Use the NFSv4 domain derived by the system
[ ] Specify a different NFSv4 domain
```

```
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.
```

```
NFSv4 Domain Name: << Value to be derived dynamically >>
```



```

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Time zones
#####
[X] Central Time - most locations
[ ] Central Time - Quintana Roo
[ ] Central Time - Campeche, Yucatan
[ ] Mexican Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas awa
[ ] US Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas near US
[ ] Mountain Time - S Baja, Mayarit, Sinaloa
[ ] Mexican Mountain Time - Chihuahua away from US border
[ ] US Mountain Time - Chihuahua near US border
[ ] Mountain Standard Time - Sonora
[ ] US Pacific Time - Baja California near US border
[ ] Mexican Pacific Time - Baja California away from US border
[ ] Mexican Central Time - Bahia de Banderas

#####
Esc-2_Continue   Esc-6_Help

```

```

> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.

Time zone: Central Time - most locations
(Mexico/General)

```

El usuario root es el que tiene el mayor nivel de privilegios y es por esto que el instalador solicita el **password de root**, el password debe definirse basándose en las buenas prácticas para la selección de contraseñas de tal forma que no pueda ser fácilmente descifrada, posteriormente para continuar tecleamos **Esc y 2** .

```

Please enter the root password for this system.

The root password may contain alphanumeric and special characters.  For
security, the password will not be displayed on the screen as you type it.

> If you do not want a root password, leave both entries blank.

Root password: *******
Root password: *******

```

Se indica que la zona va a reiniciar debido a la configuración previamente hecha, estos cambios se realizaron en **/etc/default/init**.

```
[NOTICE: Zone rebooting]

SunOS Release 5.10 Version Generic_142809 1/ 64 bit
Copyright (c) 1988, 2010, Oracle and/or its affiliates. All rights reserved.
Hostname: colaboracion-db
Reading NFS config: done.

colaboracion-db console login: Oct 31 12:31:23 colaboracion-db sendmail[6304]: My u
nqualified host name (colaboracion-db) unknown; sleeping for retry
Oct 31 12:31:23 colaboracion-db sendmail[6320]: My unqualified host name (colaborac
ion-db) unknown; sleeping for retry

colaboracion-db console login:
colaboracion-db console login: root
Password:
Oct 31 12:31:41 colaboracion-db login: ROOT LOGIN /dev/console
Last login: Sat Oct 30 14:34:58 on pts/6
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
# bash
bash-3.00#

Connected to 192.0.198.190 55-12 - aes128-cbc - hmac-md5 - rfc 83x21
```

## Zona Joomla-App

Se establece el nombre del host, para el cual se especifica joomla-app y se teclea **F2** para continuar y confirmar el nombre del Host.

```
> Confirm the following information.  If it is correct, press F2;  
to change any information, press F4.  
  
Host name: joomla-app
```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No** y posteriormente confirmando con la tecla **F2**.

```
Specify Yes if the system will use the Kerberos security mechanism.  
Specify No if this system will use standard UNIX security.  
  
Configure Kerberos Security  
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq  
[ ] Yes  
 No
```

```
> Confirm the following information.  If it is correct, press F2;  
to change any information, press F4.  
  
Configure Kerberos Security: No
```



```
> Confirm the following information. If it is correct, press F2:
to change any information, press F4.
```

```
NFSv4 Domain Name: << Value to be derived dynamically >>
```

Se debe seleccionar la zona horaria y para ello nos da diferentes formas de elegirla, la opción que se elige es la de “Continente/País/Región Geográfica”.

Se elige **Americas**, para especificar que el equipo se encuentra en América.

```
On this screen you must specify your default time zone. You can specify a
time zone in three ways: select one of the continents or oceans from the
list, select other - offset from GMT, or other - specify time zone file.
```

```
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].
```

```
Continents and Oceans
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
- [ ] Africa
x [X] Americas
x [ ] Antarctica
x [ ] Arctic Ocean
x [ ] Asia
x [ ] Atlantic Ocean
x [ ] Australia
x [ ] Europe
v [ ] Indian Ocean
```

En la sección de Country or Region, se selecciona **Mexico**.

```
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].
```

```
Countries and Regions
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
^ [ ] El Salvador
x [ ] French Guiana
x [ ] Greenland
x [ ] Grenada
x [ ] Guadeloupe
x [ ] Guatemala
x [ ] Guyana
x [ ] Haiti
x [ ] Honduras
x [ ] Jamaica
x [ ] Martinique
x [X] Mexico
v [ ] Montserrat
```

En Time Zone se indica **Central Time** para la ciudad de México, posteriormente se confirma esta zona con la tecla **F2**.

```
> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Time zones
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
[X] Central Time - most locations
[ ] Central Time - Quintana Roo
[ ] Central Time - Campeche, Yucatan
[ ] Mexican Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas awa
[ ] US Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas near US
[ ] Mountain Time - S Baja, Nayarit, Sinaloa
[ ] Mexican Mountain Time - Chihuahua away from US border
[ ] US Mountain Time - Chihuahua near US border
[ ] Mountain Standard Time - Sonora
[ ] US Pacific Time - Baja California near US border
[ ] Mexican Pacific Time - Baja California away from US border
[ ] Mexican Central Time - Bahia de Banderas
```

```
> Confirm the following information.  If it is correct, press F2;
to change any information, press F4.
```

```
Time zone: Central Time - most locations
           (Mexico/General)
```

El usuario root es el que tiene el mayor nivel de privilegios y es por esto que el instalador solicita el **password de root**, el password debe definirse basándose en las buenas prácticas para la selección de contraseñas de tal forma que no pueda ser fácilmente descifrada, posteriormente para continuar tecleamos **Esc** y **2**.



```

Please enter the root password for this system.

The root password may contain alphanumeric and special characters. For
security, the password will not be displayed on the screen as you type it.

> If you do not want a root password, leave both entries blank.

Root password: *****
Root password: *****

#####
Esc-2_Continue   Esc-6_Help

```

Se nos indica que la zona va a reiniciar debido a la configuración previamente hecha, estos cambios se realizaron en **/etc/default/init**.

```

rebooting system due to change(s) in /etc/default/init

[NOTICE: Zone rebooting]

SunOS Release 5.10 Version Generic_142909-17 64-bit
Copyright (c) 1983, 2010, Oracle and/or its affiliates. All rights reserved.

```

Finalmente que los parámetros de nombre de host, ip y el nombre sean los correctos, se verifica en **/etc/hosts**

```

bash-3.00# hostname
joomla-app
bash-3.00#
bash-3.00# ifconfig -a
lo0:5: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 ind
ex 1
    inet 127.0.0.1 netmask ffffffff
el000g0:6: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
    inet 192.9.198.213 netmask fffffff0 broadcast 192.9.198.255
bash-3.00#

```



### Zona Joomla-DB

Se establece el nombre del host, para el cual se especifica joomla-db y se teclea **F2** para continuar.

```
Enter the host name which identifies this system on the network. The name
must be unique within your domain; creating a duplicate host name will cause
problems on the network after you install Solaris.

A host name must have at least one character; it can contain letters,
digits, and minus signs (-).

Host name for e1000g0:7 joomla-db
```

Se confirma el nombre del Host con la tecla **F2**.

```
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

Host name: joomla-db
```

A continuación se configuran las políticas de seguridad, mediante la activación de **Kerberos** o de la **seguridad estándar de UNIX**. *Kerberos* es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina, mientras que la *seguridad estándar de UNIX*, no sólo abarca la autenticación sino también otros aspectos de seguridad que involucran la red, passwords, sistemas de archivos, entre otros, es por esta razón seguridad estándar de UNIX es la opción que se elige, esto realiza seleccionando **No** y posteriormente confirmando con la tecla **F2**.

```
Configure Security Policy: ~~~~~
Specify Yes if the system will use the Kerberos security mechanism.
Specify No if this system will use standard UNIX security.

Configure Kerberos Security
~~~~~
[ ] Yes
[X] No
```



```
NFS version 4 uses a domain name that is automatically derived from the
system's naming services. The derived domain name is sufficient for most
configurations. In a few cases, mounts that cross domain boundaries might
cause files to appear to be owned by "nobody" due to the lack of a common
domain name.
```

```
The current NFSv4 default domain is: ""
```

#### NFSv4 Domain Configuration

```

#####
[X] Use the NFSv4 domain derived by the system
[ ] Specify a different NFSv4 domain

```

```
> Confirm the following information. If it is correct, press F2;
to change any information, press F4.
```

```
NFSv4 Domain Name: << Value to be derived dynamically >>
```

Se debe seleccionar la zona horaria y para ello nos da diferentes formas de elegirla, la opción que se elige es la de “Continente/País/Región Geográfica”.

Se elige **Americas**, para especificar que el equipo se encuentra en América.

```

On this screen you must specify your default time zone. You can specify a
time zone in three ways: select one of the continents or oceans from the
list, select other - offset from GMT, or other - specify time zone file.

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

Continents and Oceans
#####
- [ ] Africa
x [X] Americas
x [ ] Antarctica
x [ ] Arctic Ocean
x [ ] Asia
x [ ] Atlantic Ocean
x [ ] Australia
x [ ] Europe
v [ ] Indian Ocean

#####
Esc-2_Continue Esc-6_Help

```

En la sección de Country or Region, se selecciona **Mexico**.

```

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

      Countries and Regions
      ~~~~~
^  [ ] El Salvador
x  [ ] French Guiana
x  [ ] Greenland
x  [ ] Grenada
x  [ ] Guadeloupe
x  [ ] Guatemala
x  [ ] Guyana
x  [ ] Haiti
x  [ ] Honduras
x  [ ] Jamaica
x  [ ] Martinique
x  [X] Mexico
v  [ ] Montserrat

~~~~~
Esc-2_Continue   Esc-6_Help

```

En Time Zone, se indica **Central Time** para la ciudad de México, posteriormente se confirma esta zona con la tecla **F2**.

```

> To make a selection, use the arrow keys to highlight the option and
press Return to mark it [X].

      Time zones
      ~~~~~
[X] Central Time - most locations
[ ] Central Time - Quintana Roo
[ ] Central Time - Campeche, Yucatan
[ ] Mexican Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas awa
[ ] US Central Time - Coahuila, Durango, Nuevo Leon, Tamaulipas near US
[ ] Mountain Time - S Baja, Nayarit, Sinaloa
[ ] Mexican Mountain Time - Chihuahua away from US border
[ ] US Mountain Time - Chihuahua near US border
[ ] Mountain Standard Time - Sonora
[ ] US Pacific Time - Baja California near US border
[ ] Mexican Pacific Time - Baja California away from US border
[ ] Mexican Central Time - Bahia de Banderas

~~~~~
Esc-2_Continue   Esc-6_Help

```

```

> Confirm the following information. If it is correct, press F2;
to change any information, press F4.

```

```

Time zone: Central Time - most locations
(Mexico/General)

```







---

## ANEXO 4 – Configuración de asignación de memoria mediante Memory Capping

Las zonas configuradas soportan límites de memoria física. El demonio *rcapd* y sus herramientas proveen mecanismos para la administración y configuración de los límites de memoria.

Para la configuración de los límites de memoria, se debe de contar con privilegios de “superuser”.

Primero, se debe verificar que el servicio “rcap” se encuentre habilitado usando el comando **svcs**

```
uxsop001 / # svcs | grep rcap
online          13:19:23 svc:/system/rcap:default
```

En este caso, el servicio se encuentra habilitado.

Después, para configurarle a cada zona su límite de memoria se usa el comando **zonecfg**.

Comenzaremos con la zona *colaboracion-db*.

```
uxsop001 / # zonecfg -z colaboracion-db
zonecfg:colaboracion-db> add capped-memory
zonecfg:colaboracion-db:capped-memory> set physical=1g
zonecfg:colaboracion-db:capped-memory> end
zonecfg:colaboracion-db> verify
zonecfg:colaboracion-db> commit
zonecfg:colaboracion-db> exit
```

Y se reinicia la zona.

```
uxsop001 / # zlogin colaboracion-db init 6
```

Y de esta manera la zona *colaboración-db* ya cuenta con un límite asignado de memoria física.

Para las zonas restantes es exactamente el mismo procedimiento. Para la zona *colaboracion-app* sería:

```
uxsop001 / # zonecfg -z colaboracion-app
zonecfg:colaboracion-app> add capped-memory
zonecfg:colaboracion-app:capped-memory> set physical=1g
zonecfg:colaboracion-app:capped-memory> end
zonecfg:colaboracion-app> verify
zonecfg:colaboracion-app> commit
zonecfg:colaboracion-app> exit
```

Se reinicia *colaboracion-app* usando **init 6**.

```
uxsop001 / # zlogin colaboracion-app init 6
```

Para la zona *ecommerce-db*, le asignamos 1GB de memoria física:

```
uxsop001 / # zonecfg -z ecommerce-db
zonecfg:ecommerce-db> add capped-memory
zonecfg:ecommerce-db:capped-memory> set physical=1g
zonecfg:ecommerce-db:capped-memory> end
zonecfg:ecommerce-db> verify
zonecfg:ecommerce-db> commit
zonecfg:ecommerce-db> exit
```

Reiniciamos la zona:

```
uxsop001 / # zlogin ecommerce-db init 6
```

A la zona *ecommerce-app* le asignamos 2GB:

```
uxsop001 / # zonecfg -z ecommerce-app
zonecfg:ecommerce-app> add capped-memory
zonecfg:ecommerce-app:capped-memory> set physical=2g
zonecfg:ecommerce-app:capped-memory> end
zonecfg:ecommerce-app> verify
zonecfg:ecommerce-app> commit
zonecfg:ecommerce-app> exit
```

Igualmente, reiniciamos la zona *ecommerce-app*:

```
uxsop001 / # zlogin ecommerce-app init 6
```

A la zona *joomla-db* se le asigna 1GB de memoria:

```
uxsop001 / # zonecfg -z joomla-db
zonecfg:joomla-db> add capped-memory
zonecfg:joomla-db:capped-memory> set physical=1g
zonecfg:joomla-db:capped-memory> end
zonecfg:joomla-db> verify
zonecfg:joomla-db> commit
zonecfg:joomla-db> exit
```

Reiniciamos la zona:

```
uxsop001 / # zlogin joomla-db init 6
```

Configurando la última zona, *joomla-app*, le asignamos 1GB de memoria física:

```
uxsop001 / # zonecfg -z joomla-app
zonecfg:joomla-app> add capped-memory
zonecfg:joomla-app:capped-memory> set physical=1g
zonecfg:joomla-app:capped-memory> end
zonecfg:joomla-app> verify
zonecfg:joomla-app> commit
zonecfg:joomla-app> exit
```

Y por último, reiniciamos la zona:

```
uxsop001 / # zlogin joomla-app init 6
```

Usando el commando ***rcapstat*** podemos verificar y monitorear los límites de memoria asignados a cada zona:

```
uxsop001 / # rcapstat -z 5 3
  id zone          nproc    vm    rss    cap      at avgat    pg avgpg
  25 colaboracion-db -    206M  259M 1024M    0K   0K     0K   0K
  26 colaboracion-app -    251M  259M 1024M    0K   0K     0K   0K
  27 ecommerce-db    -    206M  261M 1024M    0K   0K     0K   0K
  28 ecommerce-app   -    255M  271M 2048M    0K   0K     0K   0K
  29 joomla-db       -    205M  257M 1024M    0K   0K     0K   0K
  30 joomla-app      -    245M  256M 1024M    0K   0K     0K   0K
  id zone          nproc    vm    rss    cap      at avgat    pg avgpg
  25 colaboracion-db -    206M  259M 1024M    0K   0K     0K   0K
  26 colaboracion-app -    251M  259M 1024M    0K   0K     0K   0K
  27 ecommerce-db    -    206M  261M 1024M    0K   0K     0K   0K
  28 ecommerce-app   -    255M  271M 2048M    0K   0K     0K   0K
  29 joomla-db       -    205M  257M 1024M    0K   0K     0K   0K
  30 joomla-app      -    245M  256M 1024M    0K   0K     0K   0K
  id zone          nproc    vm    rss    cap      at avgat    pg avgpg
  25 colaboracion-db -    206M  259M 1024M    0K   0K     0K   0K
  26 colaboracion-app -    251M  259M 1024M    0K   0K     0K   0K
  27 ecommerce-db    -    206M  261M 1024M    0K   0K     0K   0K
  28 ecommerce-app   -    255M  271M 2048M    0K   0K     0K   0K
  29 joomla-db       -    205M  257M 1024M    0K   0K     0K   0K
  30 joomla-app      -    245M  256M 1024M    0K   0K     0K   0K
```

En la sintaxis del comando, el “5” especifica el intervalo de tiempo en segundos para mostrar el informe mientras que el “3” regresa 3 muestras en ese intervalo.