

## **Colegio de Ciencia y Tecnología**

# **“SISTEMA DE RESPALDOS DE UN CENTRO DE DATOS DE CLIMA ESPACIAL CON UNA ARQUITECTURA CLIENTE SERVIDOR”**

## **TESIS**

Que para obtener el título de:

**Licenciada en Ingeniería de Software**

Presenta:

**Angélica María Villagómez Cabral**

Director:

**M. en C. Enrique Cruz Martínez**

Codirector:

**Dr. Víctor Hugo de la Luz Rodríguez**

Ciudad de México, julio de 2020

## SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



## UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

### RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

### DERECHOS RESERVADOS<sup>©</sup>

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.



# Dedicatoria

A mis padres Ana María Cabral García, Manuel Villagómez Trejo †  
por la educación que me dieron, la paciencia y la confianza que me  
han tenido, por creer en mí, apoyarme en todas y cada una de mis  
metas.

A mis hermanos Ana Alicia Villagómez Cabral, Octavio Villagómez  
Cabral y Manuel Villagómez Cabral, por cada uno de sus consejos y  
cariño que me han dado.

A mis cuñadas y sobrino(a)s por el cariño que me han brindado.

Nuestra recompensa se encuentra  
en el esfuerzo y no en el resultado.

Un esfuerzo total  
es una victoria completa.

*Mahatma Gandhi*



# Agradecimientos

A mi director de tesis el M. en C. Enrique Cruz Martínez y a mi Codirector Dr. Víctor Hugo De la Luz Rodríguez por haber aceptado dirigir este trabajo, además de esta tesis me brindaron apoyo para ser partícipe en el proyecto CONACYT Ciencia Básica (254497), Repositorios Institucionales (268273).

A la Universidad Autónoma de la Ciudad de México (UACM), por sus enseñanzas, pero sobre todo por darme la oportunidad de obtener una carrera.

Al Laboratorio de Cómputo para la Enseñanza de las Ciencias (LACECI), por abrirme sus puertas para tener un lugar, en el cual poder realizar tareas y proyectos durante el desarrollo de mi carrera, así como a cada uno de mis amigos por haber compartido conmigo sus conocimientos, por esos buenos momentos y grandes aprendizajes que me han enseñado a convivir y poder trabajar en equipo.

A el M. en C. Fernando Robles Morales, por todo su apoyo que me brindó en el desarrollo de este proyecto y el conocimiento que me ha compartido durante el desarrollo de mi carrera.

A mis lectores M. en Adm. de TI. Juan Carlos Cruz Romero, M. en I. Víctor Manuel Macías Medrano, M. en C. Agustín González Villanueva.

A Moisés Alvarado Hermida, por el apoyo que me brindó en el desarrollo de mi tesis.

A mis amigas Yulli E. Aguillón Martínez, Sandy B. Vallejo González por siempre estar a mi lado, regalándome consejos para seguir adelante.



# Índice general

Índice de Figuras	IX
Índice de Cuadros	XI
Resumen	1
<b>1. Sistemas de respaldos</b>	<b>3</b>
1.1. Introducción . . . . .	3
1.2. Sistemas de respaldos . . . . .	4
1.3. Sistema de almacenamiento . . . . .	5
1.4. Protocolos de comunicación en red para respaldos . . . . .	6
1.5. Arquitectura . . . . .	7
1.6. Políticas de respaldos . . . . .	11
1.7. Respaldo de datos del clima espacial . . . . .	14
<b>2. Diseño del Sistema de Respaldos</b>	<b>17</b>
2.1. Planteamiento del problema . . . . .	17
2.2. Análisis de requisitos . . . . .	19
2.3. Diagramas . . . . .	20
2.4. Diseño del sistema de respaldos . . . . .	22
2.5. Plan de recuperación de datos . . . . .	22
2.5.1. Tolerancia a fallas . . . . .	25
2.5.2. Recuperación de datos . . . . .	25
<b>3. Implementación del sistema de respaldos</b>	<b>27</b>
3.1. Configuración del hardware del servidor . . . . .	27
3.2. Configuración del software del sistema . . . . .	31
3.3. Recuperación de datos . . . . .	34
3.4. Pruebas de respaldo de la información . . . . .	36
3.5. Pruebas de recuperación de información . . . . .	38

<b>4. Análisis</b>	<b>41</b>
4.1. Rendimiento de lectura y escritura . . . . .	41
4.2. Análisis de pruebas de respaldo . . . . .	48
4.3. Proyectos futuros . . . . .	51
<b>5. Conclusiones</b>	<b>53</b>
<b>A. Apéndice</b>	<b>55</b>
A.1. Creación de gráficas con octave . . . . .	55
<b>Bibliografía</b>	<b>58</b>

# Índice de figuras

1.1.	Modelo Cliente-Servidor . . . . .	8
1.2.	Modelo Peer To Peer . . . . .	10
1.3.	Diagrama del servicio de clima espacial desde el 2016 . . . . .	15
2.1.	Radiotelescopio Callisto . . . . .	18
2.2.	Arreglo de antenas dipolares Mexart . . . . .	18
2.3.	Diagrama de secuencia Callisto . . . . .	20
2.4.	Diagrama de secuencia MEXART . . . . .	20
2.5.	Diagrama de secuencia sistema de respaldo . . . . .	21
2.6.	Diagrama de secuencia recuperación de datos . . . . .	21
2.7.	Sistema de respaldo . . . . .	22
2.8.	Página Callisto . . . . .	23
2.9.	Página Mexart . . . . .	24
3.1.	Servidor Huitzilopochtli UACM . . . . .	28
3.2.	Particiones de Servidor Huitzilopochtli . . . . .	30
3.3.	Configuración de Sistema Operativo CentOS7 . . . . .	31
3.4.	Crear usuario . . . . .	35
3.5.	Crear contraseña . . . . .	35
3.6.	Configuración firewall . . . . .	35
4.1.	Archivo vacío 14 kb . . . . .	42
4.2.	Archivo vacío 254 kb . . . . .	43
4.3.	Archivo vacío 1 Mb . . . . .	44
4.4.	Archivo random 14 kb . . . . .	45
4.5.	Archivo random 254 kb . . . . .	46
4.6.	Archivo random 1 Mb . . . . .	47
4.7.	Callisto vs Mexart . . . . .	48
4.8.	Callisto vs Mexart . . . . .	50



# Índice de cuadros

2.1. Requerimientos funcionales. . . . .	19
2.2. Requerimientos no funcionales. . . . .	19
4.1. Archivo vacío 14 kb . . . . .	42
4.2. Archivo vacío 254 kb . . . . .	43
4.3. Archivo vacío 1 Mb . . . . .	44
4.4. Archivo random 14 kb . . . . .	45
4.5. Archivo random 254 kb . . . . .	46
4.6. Archivo random 1 Mb . . . . .	47
4.7. Métrica Callisto megabyte/Segundos Transferencia . . .	48
4.8. Métrica Callisto vs Mexart megabyte/clock time Trans- ferencia . . . . .	49



# Resumen

Este trabajo forma ya parte del Repositorio Institucional de Clima Espacial de la Universidad Nacional Autónoma de México (UNAM), del cual la Universidad Autónoma de la Ciudad de México (UACM) forma parte a partir de éste trabajo de Tesis.

El Repositorio Institucional de Clima Espacial (RICE) además de tener una red de instrumentación amplia, procesa y almacena señales en forma digital; para esto se desarrolló un Centro de Datos que pueda contener la información sobre todos los aparatos, conexión con redes mundiales de monitoreo del Sol y simulaciones numéricas sobre fenómenos solares que afectan a la ionosfera.

El Centro de Datos para RICE, será realizado mediante software libre, esto con la finalidad de que el sistema se adapte a las necesidades que tenga el servidor.

Se miden con métricas básicas la Lectura/Escritura en el arreglo de discos y el desempeño de la Red con el ancho de banda en la transferencia de archivos.



# Capítulo 1

## Sistemas de respaldos

### 1.1. Introducción

La contribución de este trabajo con la Universidad Nacional Autónoma de México (UNAM) es llevar acabo un sistema de respaldo *backup* donde la información que ellos tienen almacenada en su servidor será respaldada en el servidor que nos brindó la misma Universidad.

Se desarrolló el sistema backup [1] que consiste en la configuración del servidor con 1 ó 2 procesadores y un arreglo de 5 hasta 20 discos duros de un Tb en configuración de RAID 5, scripts de respaldos automáticos y el uso de un ancho de banda semi-dedicado con tolerancia a fallas.

La implementación de un sistema de respaldos de clima espacial sirve como alternativa ante una potencial pérdida de datos por diferentes eventos.

En está tesis, se describirán las principales componentes de un sistema de respaldos y se pusieron en marcha utilizando software libre y una metodología cliente servidor además de que esto se aplicara sobre el servidor que brindó la UNAM.

## 1.2. Sistemas de respaldos

El respaldo y recuperación permite que las Instituciones tengan vigente su información en caso de haber tenido alguna falla en su sistema de software o en el hardware. La información puede ser recuperada a través de una copia de respaldo, está depende de qué tanta información se encuentra en el sistema, la cual se podría hacer por minuto, hora, día, mes o años.

Para realizar un respaldo de un sistema especializado que permita hacer copias con información no redundante, para guardar lo que se ha modificado y no toda la información nuevamente se utilizará un respaldo incremental, una de sus ventajas es que se realiza rápido.

Como ya se sabe la información es el activo más importante con el que se puede contar en las universidades, proyectos estudiantiles, así como empresariales, tomando en cuenta que hay información confidencial y datos personales. El sistema de respaldos debe estar siempre disponible para recuperar la información para mantenerla segura y evitar información redundante.

Algunos de los sistemas de respaldos con los que se puede contar son:

*Linear Tape-Open (LTO)* es una tecnología de cinta magnética de almacenamiento de datos, como alternativa de estándares abiertos a los formatos de cinta magnética patentada.

*Data archiving* archivo de datos, el archivado de datos se almacena de tal forma que algunas empresas se encargan de inventariar sus datos e identificar cuales son candidatos para el archivo ya que a estos se les podrá poner solo lectura.

*solid-state drive (SSD) y disco duro portátil (HDD)* estas unidades de discos son fáciles de transportar de una computadora a otra, estos son utilizados para almacenamiento de información de una Computadora hasta podría realizarse una copia de seguridad de un Sistema Operativo.

### 1.3. Sistema de almacenamiento

Los sistemas de almacenamiento, son sistemas para la gestión, organización y administración de información desarrollados con software libre ó bajo alguna licencia. De los cuales en este proyecto se usaron los siguientes:

*Bacula* es una colección de herramientas de respaldo capaz de cubrir las necesidades de equipos bajo redes IP. Se basa en una arquitectura Cliente-servidor que resulta eficaz y fácil de manejar, dada la amplia gama de funciones y características que brinda; copiar y restaurar ficheros dañados o perdidos.

Además, debido a su desarrollo y estructura modular, Bacula se adapta tanto al uso personal como profesional, desde un equipo hasta grandes parques de servidores.

*Tivoli Storage Manager (TSM)* es un software que parte de la serie IBM Total Storage, centralizado y basado en políticas que permite la administración de los recursos de almacenamiento, como copia de seguridad de datos.

*Veeam* es una empresa privada del sector de la tecnología de la información que desarrolla software de gestión de la virtualización, copia de seguridad (backup) y plan de recuperación ante desastres (Disaster Recovery) para entornos virtuales de VMware.

*EaseUS Todo Backup* es un sistema de version gratuita, aunque es para usuarios domésticos, si se requiriera para algo más sofisticado se necesita pagar licencia. Además, de que realiza las copias en la nube, utilizando el almacenamiento en DropBox, One Drive, o Google Drive manteniendo su información de forma segura en caso de que aumente el almacenamiento deberá pagar por más espacio de almacenamiento.

## 1.4. Protocolos de comunicación en red para respaldos

Es una regla o estándar que permite la conexión, comunicación y transferencia entre dos nodos. Pueden ser implementados por Hardware, Software o una combinación de ambos, para poder respaldar información.

**Web server o servidor web**, Es el encargado de procesar una aplicación del lado del servidor, realizando conexiones unidireccionales y bidireccionales, así como síncronas y asíncronas, este es utilizado mediante el protocolo HTTP, que pertenece al modelo Open System Interconnection (OSI).

**Protocolo FTP**, File Transfer Protocol, Protocolo de Transferencia de Archivos, es un protocolo de red para la transferencia de archivos entre sistemas interconectados o enlazados a Internet, basado en la arquitectura cliente-servidor.

**Secure SHell (SSH)**, es un protocolo que facilita la comunicación entre dos sistemas utilizando una arquitectura cliente servidor, permitiendo a los usuarios conectarse a un host *es el servidor remoto al que se esta intentando acceder*, una de sus ventajas de SSH es que encripta la sesión de conexión.

La instrucción SSH le indica a tu sistema que desea abrir una Conexión de Shell Segura y cifrada.

La ventaja del utilizar SSH es el uso del cifrado para tener una transferencia segura de información entre el host y el servidor.

Hay tres tecnologías de cifrado diferentes utilizadas por SSH:

- El cifrado simétrico. En el cifrado simétrico se genera una clave secreta llamada clave compartida (shared key), que va a servir tanto para el cifrado como el descifrado, tanto por el cliente como por el host, se utilizan para cifrar toda la comunicación durante

la sesión SSH.

- Cifrado asimétrico. Utiliza dos claves separadas para el cifrado y el descifrado, la relación entre las dos claves es altamente compleja: un mensaje cifrado por la clave pública de una máquina sólo puede ser descifrado por la misma clave privada de la máquina.

Esta relación unidireccional, significa que la clave pública no puede descifrar sus propios mensajes ni descifrar nada cifrado por la clave privada.

El cifrado asimétrico, no se utiliza para cifrar toda la sesión SSH sólo se utiliza durante el algoritmo de intercambio de claves de cifrado simétrico. Antes de iniciar una conexión segura, ambas partes generan pares de claves públicas-privadas temporales las cuales se comparten para producir la clave secreta compartida.

- Hashing. Este cifrado es unidireccional, este difiere de los dos anteriores debido a que estas nunca están destinadas a ser descifradas, generan un valor único de una longitud fija.

## 1.5. Arquitectura

La arquitectura de un sistema es el diseño de más alto nivel de su estructura, especificados por separado. Las principales características, son que el sistema sea fiable, manejable, adaptable y rentable.

- **Modelo Cliente-Servidor**, en el modelo cliente servidor [2] se realizan diversas tareas donde los procesos del cliente van a interactuar con los procesos del servidor, todo esto para acceder a los recursos compartidos que administran.

*Cliente* es el que solicita un servicio específico en un sistema de tipo servidor, los clientes centralizan diferentes aplicacio-

nes y recursos en el servidor.

El *Servidor* es una máquina la cual va a comunicar con otra máquina llamada cliente donde ésta le va proporcionar el servicio apropiado al cliente ver figura 1.1 [2] .

- Ventajas.
- Centralización de control.
- Escalabilidad.
- Fácil mantenimiento.
- Tecnologías maduras y robustas.

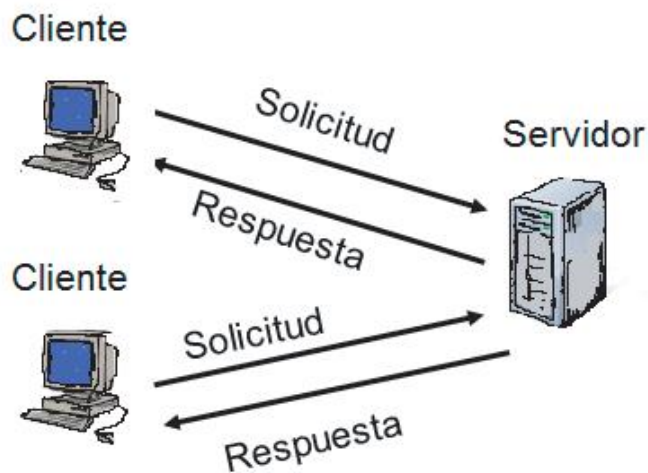


Figura 1.1: Modelo Cliente-Servidor

En este modelo podemos tener diversas configuraciones como son: tener más de un cliente conectado al servidor, también se podría tener un grupo de servidores interconectados para dar servicio a un grupo de clientes, ver figura 1.1.

Este modelo es el que más se ajusta con el sistema de respaldo debido a que para realizar el respaldo requerimos del cliente que es la máquina a donde se está haciendo el respaldo y el servidor

es a quien le pedimos la información, por lo tanto, sin este modelo no podríamos realizar el respaldo ver figura 1.1.

- **Arquitectura en tres capas** ésta arquitectura es usada en la gran mayoría de sistemas. Se suele usar en sistemas que implementan un modelo de negocio, como sería una tienda online, aplicación para la gestión de datos.

Lo que hace la arquitectura en tres capas, es que cada capa solo se comunica con la capa inferior; estas tres capas son:

- **Persistencia:** Esta capa se encarga de guardar los datos. Será donde se gestione todo lo relativo a la base de datos y a la creación, edición y borrado de datos de ésta.
- **Negocio:** En esta capa se gestiona la lógica de la aplicación. Es donde se dice que se hace con los datos. Por ejemplo para una aplicación de gestión de una biblioteca será donde se gestione cuantos préstamos puede tener un usuario, que ocurre si un usuario se retrasa al devolver un libro. Estará conectada con la capa de persistencia para poder realizar sus funciones.
- **Presentación:** En esta capa se crea la interfaz del usuario. Su única función es pasarle las acciones que realice el usuario a la capa de negocio.

- **Peer-to-Peer.**

Una red peer-to-peer, red de pares (P2P), este es un caso muy interesante pues cada peer puede tomar el papel tanto del cliente como del servidor, una de sus ventajas del sistema peer-to-peer es que todos los recursos disponibles son proporcionados por los peer.

Al momento de necesitar distribuir algún contenido los peers aportan sus recursos para transmitirlo a los demás peers, cuando se agrega un nuevo peer la demanda incrementa pero la capacidad general del sistema también, todo esto no es posible en el modelo cliente servidor ver figura 1.2 [2].

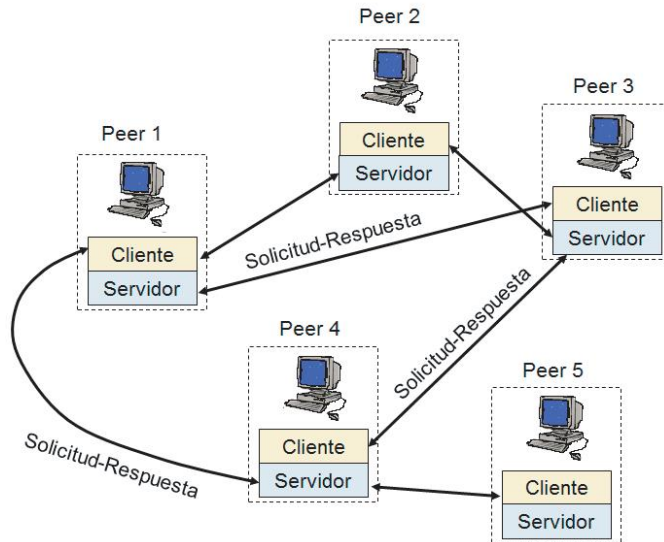


Figura 1.2: Modelo Peer To Peer

Ventajas de peer to peer.

- Los nodos comparten recursos
  - Se pueden ejecutar algoritmos distribuidos
  - Flexibilidad
  - Mayor robustez del sistema
- **Modelo vista controlador** mejor conocido como modelo MVC, que divide una aplicación interactiva en 3 partes.
- **Modelo:** Contiene la funcionalidad y los datos básicos
  - **Vista:** Muestra la información al usuario (se puede definir más de una vista)

- **Controlador:** Maneja la entrada del usuario

Algunas de las ventajas de utilizar este modelo es para tener un mejor control en el código y hacer más eficiente algunas búsquedas para realizar una modificación. Es más común encontrarlo en aplicaciones World Wide Web en los principales lenguajes de programación.

## 1.6. Políticas de respaldos

Son criterios de bajo nivel establecidos dentro del Software de Respaldo, se refiere a las reglas que controlan el respaldo donde se garantice la continuidad de la empresa, para ello se deben realizar los backups y se debe definir el procedimiento donde se revisa: de que hacer copia, el tipo de copia, la periodicidad, la vigencia, así como llevar el control de los soportes y vigilar que solo acceda el personal autorizado.

- **Respaldo total:** Respaldo de todo el sistema de archivos (full dump). Este respaldo lo que realiza es muy sencillo debido a que guarda toda la información que esté en el sistema de origen y se almacena totalmente en el sistema de respaldo [3] .

Al realizar estos respaldos totales podemos observar que pueden ser muy costosos en cuanto al tiempo en que se realiza el mismo y también que ocuparía todo el espacio que se pueda tener disponible en ese momento.

Cuando hablamos de un respaldo completo es porque guardan desde el sistema operativo de la máquina o la estructura de directorios hasta los logs, perfiles de usuario, estructuras de datos y toda aquella información necesaria. Si restauramos la copia en una máquina que no sea la de origen, pueda funcionar exactamente de la misma manera que en el momento en que se hizo la copia de seguridad.

Para realizar la copia se podrá estar usando un software de copias de seguridad o bien algún mandato u opción proporcionada por el propio sistema operativo.

Este tipo de respaldo permite, en caso de una catástrofe realizar una rápida restauración de los archivos, pues volcando el último respaldo se tiene toda la información completa.

- **Respaldo incremental:** Se respaldan los archivos modificados luego del último respaldo total. Esta política de respaldo disminuye el tiempo que emplea el sistema en realizar el respaldo. Ante la restauración del sistema se debe volcar primero el respaldo total y al último respaldo incremental.

El software que se utilice para realizar este tipo de respaldos requiere realizar la validación de los bits de modificación desde el primer archivo, que este bit está básicamente asociado a la memoria de la computadora o del servidor.

Cuando se realiza el respaldo total se desactiva este bit, después de ahí cada archivo que se modifique se activará y se verá reflejado en los bits de modificación, cuando estos sean guardados se desactivara nuevamente este bit, de tal forma que solo estos archivos son los que se guardaran en el respaldo incremental.

- **Respaldo diferencial:** Se realizan respaldos por día, el primero será luego del último total, el siguiente será luego del último diferencial. Al respaldar solo los archivos modificados durante ese día se reduce el tiempo de respaldo, como contrapartida ante una restauración se debe volcar primero el respaldo total, y luego en forma ordenada todos los respaldos diferenciales, con esto se aumenta el tiempo de respuesta ante una falla.
- **Respaldo en línea** las copias de seguridad en línea se toman en tiempo real y proporcionan una alta redundancia y tolerancia a

fallas.

Estos mecanismos de respaldo en línea son necesarios en sistemas muy críticos donde no puede permitirse el lujo de perder ni siquiera una pequeña fracción de datos, los centros de datos que brindan soporte a varias organizaciones donde diferentes partes del mismo trabajo están siendo realizadas por diferentes centros.

- **Respaldo de línea cercana** las copias de seguridad cercanas a la línea son las copias de seguridad realizadas casi en tiempo real, pero no en tiempo real. Hay una brecha entre el sistema actual (estado de datos) y en el dispositivo de respaldo.

El concepto de salto electrónico puede permitir esto, si la transferencia de datos por lotes a otro sistema o sistema alternativo externo se puede llevar a cabo con frecuencia.

- **Respaldo sin conexión** las copias de seguridad sin conexión son la forma más común de copia de seguridad utilizada. Estas copias de seguridad se hacen en cintas o discos duros externos u otros medios según sea relevante.

Se puede hacer una copia de seguridad de los datos del sistema individual en discos duros externos. servidores y otros sistemas con una gran cantidad de datos almacenados en ellos también se pueden hacer copias de seguridad en cintas utilizando bibliotecas de cintas automatizadas en las que se guardan las copias programadas.

A veces, las copias de seguridad realizadas cuando los sistemas están fuera de línea (no se utilizan) se consideran copias de seguridad fuera de línea, como una copia de seguridad realizada cuando no se realizaba ninguna transacción en la base de datos o la base de datos estaba en modo fuera de línea.

## 1.7. Respaldo de datos del clima espacial

El monitoreo del clima espacial es muy importante debido a que las tormentas solares pueden producir tres fenómenos: emisión de radiación electromagnética ó fulguraciones solares, flujos de partículas muy energéticas en el viento solar que alcanzan la magnetósfera terrestre que pueden presentar peligros por radiación para naves espaciales, comerciales, astronautas y Eyecciones de Masa Coronal (CME), las cuales expulsan un gas incandescente con velocidades que pueden rebasar los 2000 km/s y llegar a la Tierra en un día.

Estos eventos solares generan fenómenos naturales como son las auroras boreales. También nos afecta en cuestión de la tecnología como es en la interferencia de las señales de los sistemas de telecomunicaciones y pueden llegar a afectar las redes de distribución eléctrica.

El 27 de marzo del 2016, se oficializó el SCiESMEX de la Universidad Nacional Autónoma de México <http://www.sciesmex.unam.mx/> [4], a través del Instituto de Geofísica con una red de instrumentos para monitorear la actividad del Sol y enviar alertas.

Como podemos observar en el siguiente diagrama, podemos ver que se ha formado con ayuda de diversas asociaciones de gobierno, así como de la Universidad Autónoma de la Ciudad de México (UACM). (figura 1.3).

Un proyecto asociado al SCiESMEX de vital importancia es el Repositorio Institucional de Clima Espacial (RICE) [5] el cual es el encargado del almacenamiento de datos en tiempo real de la red de instrumentación de clima espacial del país.

En este trabajo de tesis se describe la implementación de un sistema de respaldos para el RICE, que conforma la primera fase para que el Laboratorio de Computo para la Enseñanza de las Ciencias (LACECI) sea un repositorio de datos secundario de RICE [5].

En los siguientes capítulos de este trabajo se describe el diseño e implementación de este sistema de respaldos.

DIAGRAMA DE OPERACIÓN

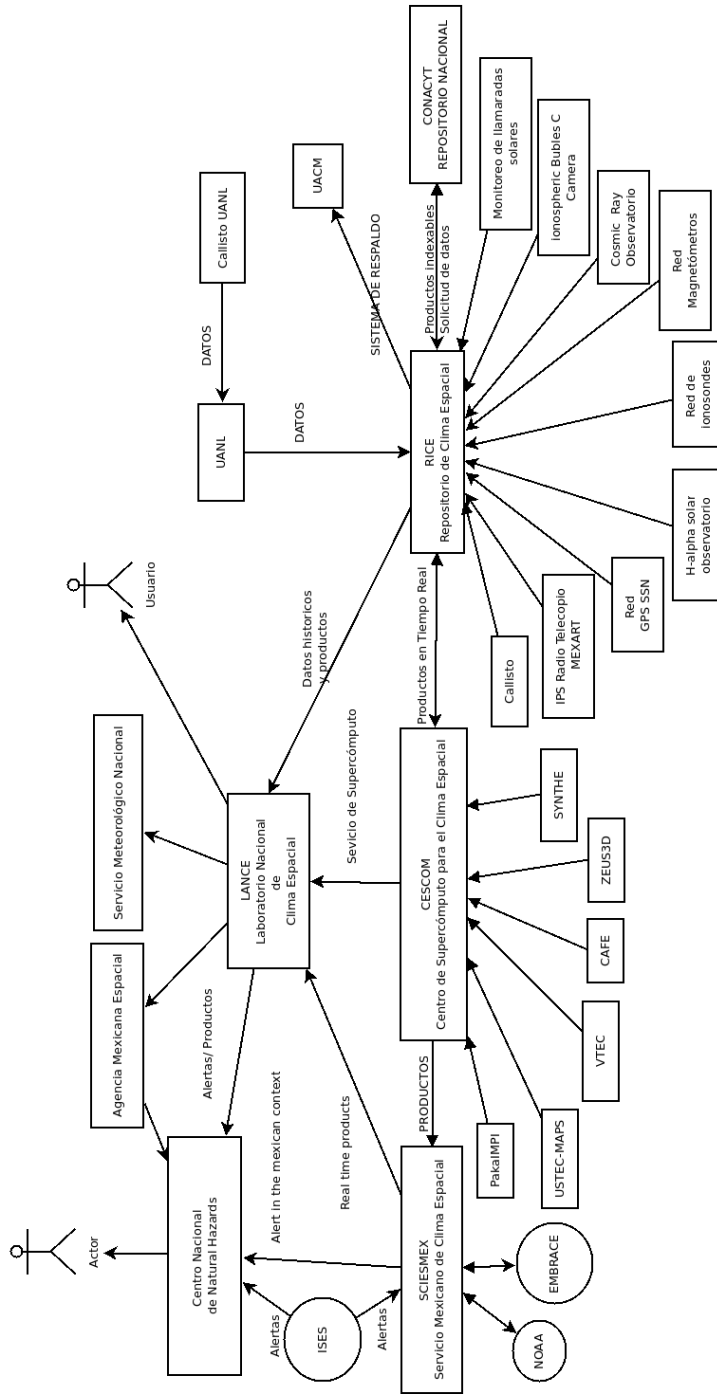


Figura 1.3: Diagrama del servicio de clima espacial desde el 2016



# Capítulo 2

## Diseño del Sistema de Respaldos

### 2.1. Planteamiento del problema

Actualmente en México existe un grupo que monitorea los eventos del Clima Espacial llamado Servicio de Clima Espacial – (SCiESMEX) dirigido por el Instituto de Geofísica (IGUM) de la Universidad Nacional Autónoma de México (UNAM) en la unidad Morelia Michoacán (<http://www.sciesmex.unam.mx/>) [4].

El *Radiotelescopio Callisto* y el *arreglo de antenas* (tipo dipolo) están ubicados en el Observatorio de Centelleo Interplanetario, Mexican Array Radio Telescope (MEXART) en Coeneo, Michoacán.

Actualmente la red de instrumentación del servicio de clima espacial mexicano, está creciendo en aparatos de medición a lo largo de la República Mexicana. A continuación se describen los aparatos que tienen más tiempo en operación y qué se encuentran en Michoacán.

- **El Radiotelescopio Callisto** está conformado por una antena yagi, un amplificador de bajo ruido y el receptor Callisto, puede operar en una banda ancha sintonizable entre 50 y 800 MHz. El Radiotelescopio hace la recolección de datos, de dónde finalmente se capta la señal de la antena digitalmente en una computadora y los datos se envían a través de Internet ver figura 2.1 [6].



Figura 2.1: Radiotelescopio Callisto

- **Mexart** consiste en un arreglo de 4096 ( $64 \times 64$ ) antenas (tipo dipolo) cubriendo un área física total de 9500 metros cuadrados. En la dirección este-oeste el arreglo tiene una longitud de 140 metros y en la dirección norte-sur el arreglo mide 80 metros ver figura 2.2 [7].



Figura 2.2: Arreglo de antenas dipolares Mexart

## 2.2. Análisis de requisitos

A continuación se en listan los requisitos funcionales y no funcionales, estos son necesarios para identificar el alcance del sistema que se va a desarrollar.

Núm RF	Descripción
RF1	El sistema debe respaldar la información por día.
RF2	El sistema debe generar las carpetas necesarias para su almacenamiento de la información.
RF3	El sistema debe activarse en automático, aún cuando se recupere de un reinicio el equipo.
RF4	El sistema debe estar activo en todo momento para la recuperación de datos.
RF5	El sistema debe tener un usuario para quien recupere la información.

Cuadro 2.1: Requerimientos funcionales.

Núm RF	Descripción
RNF1	Debe ser desarrollado con Software Libre.
RNF2	Tiempos de respuesta cortos.
RNF3	Seguridad en la cuenta de usuario.
RNF4	Organización en el almacenamiento de datos.

Cuadro 2.2: Requerimientos no funcionales.

## 2.3. Diagramas

Diagrama de secuencia Callisto

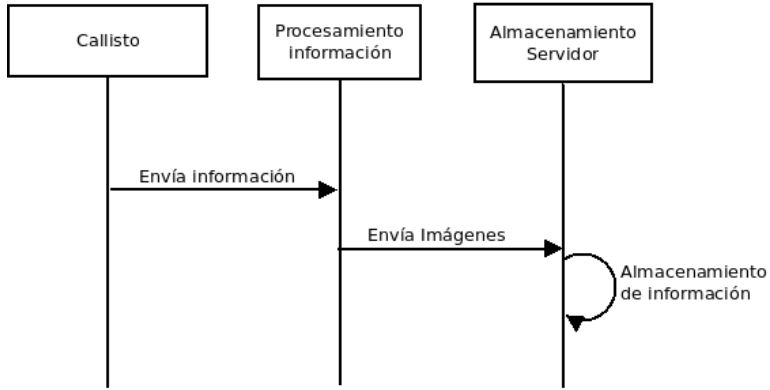


Figura 2.3: Diagrama de secuencia Callisto

Diagrama de secuencia MEXART

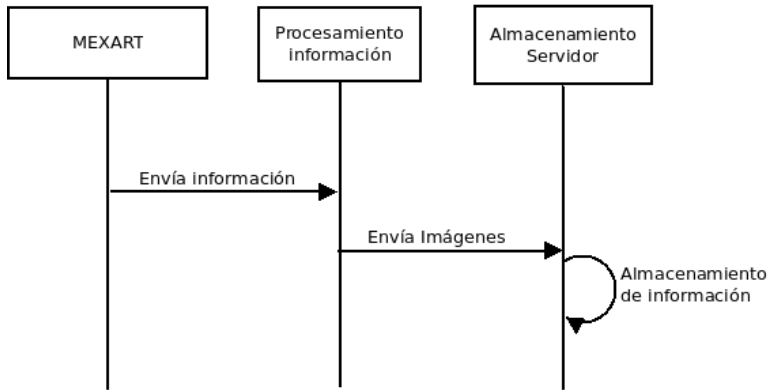


Figura 2.4: Diagrama de secuencia MEXART

Diagrama de secuencia Sistema de Respaldo

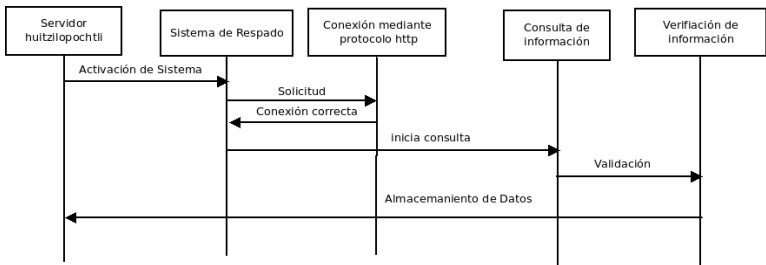


Figura 2.5: Diagrama de secuencia sistema de respaldo

Diagrama de Secuencia Recuperación de Datos

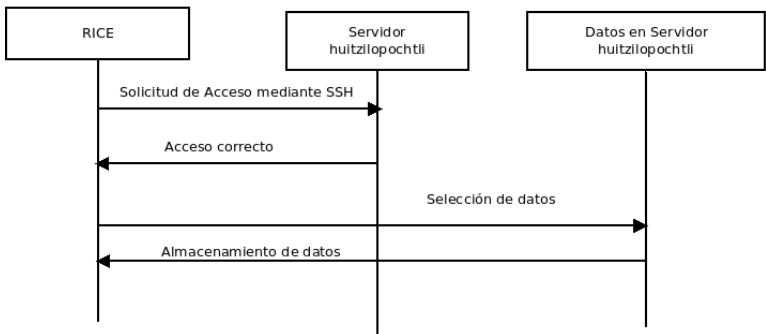


Figura 2.6: Diagrama de secuencia recuperación de datos

## 2.4. Diseño del sistema de respaldos

**Software libre (SL)**<sup>1</sup>, es la libertad de ejecutar, copiar, distribuir, estudiar y modificar, el código fuente para mejorar el software, también da la libertad de distribuir el software para ayudar a terceros. El acceso al código fuente es una condición necesaria para ello.

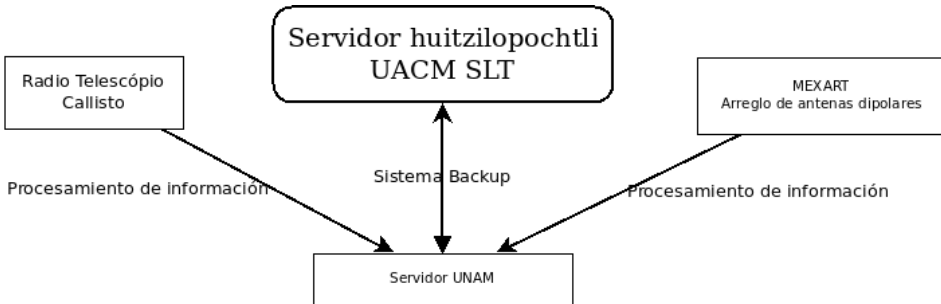


Figura 2.7: Sistema de respaldo

En la figura 2.7 se muestra la vinculación entre el proyecto global y el Repositorio Internacional de Clima Espacial (RICE), el Repositorio Nacional.

Para esta configuración se trabajará sobre el sistema operativo CentOS 7<sup>2</sup>, en base a un shell (bash), la ventaja de utilizar bash es que no hay necesidad de instalarlo pues ya es parte del mismo sistema.

## 2.5. Plan de recuperación de datos

Los archivos se van a almacenar generando una estructura de directorios de la siguiente manera:

Primero se va a crear un directorio el cual su nombre será el año actual, dentro de ese directorio se genera otro con el mes y finalmente

---

<sup>1</sup>El término *SL* se le atribuye al estadounidense Richard Stallman, quien fue fundador de la Free Software Foundation (Fundación del Software Libre.)

<sup>2</sup>CentOS es una bifurcación a nivel binario de la distribución GNU/Linux Red Hat Enterprise Linux RHEL

dentro de él se generara otra carpeta con el número del día donde se va a realizar el respaldo.

Este respaldo se va a generar cada 24 horas, debido a que se respaldan datos en tres tiempos distintos en el Servidor principal de Morelia Michoacán.

La primera página de la cual se va respaldar información es: [www.rice.unam.mx/callisto](http://www.rice.unam.mx/callisto) (ver figura 2.8), esta página almacena imágenes cada 15 minutos. Aunque se generan cerca de 15x4x24 archivos por día, el tipo de datos que se almacenan son archivos tif (TIF es un formato de archivo informático para almacenar imágenes de mapa de bits), se realizara el corte cada 24 hrs para respaldar todo lo generado en ese lapso de tiempo.

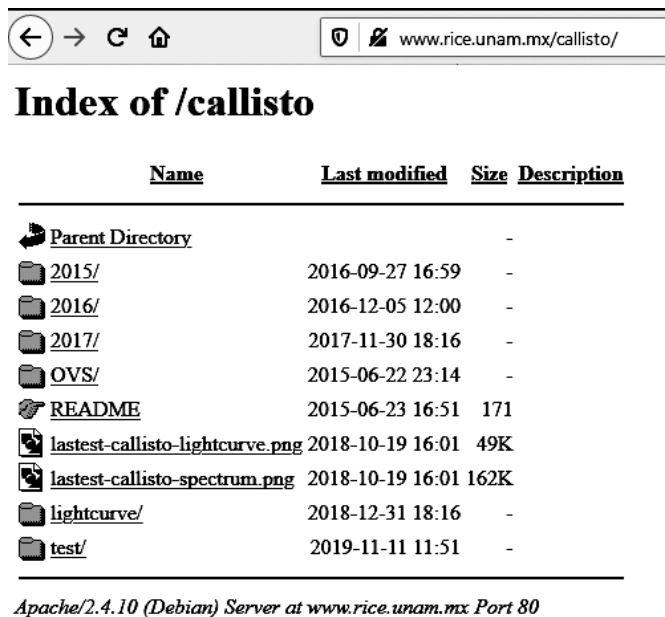


Figura 2.8: Página Callisto

La segunda página es [www.rice.unam.mx/mexart/ch00](http://www.rice.unam.mx/mexart/ch00) (ver figura de del sitio 2.9), en ella se almacenan imágenes cada 24 horas formato UTC + 2 horas. En este sitio se almacena un archivo por día que aproximadamente tienen tamaño de 45 a 50 megabytes obtenidas de las mediciones de un conjunto de antenas dipolares.

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
<a href="#">Parent Directory</a>		-	
<a href="#">2017/</a>	2017-12-05 19:10	-	
<a href="#">2018/</a>	2018-12-02 19:43	-	
<a href="#">2019-08-06-chan-0.txt.gz</a>	2019-08-07 19:10	7.6M	
<a href="#">2019/</a>	2019-11-02 19:10	-	
<a href="#">mexart-2018-10-25-chan-00.dat</a>	2018-10-25 19:05	33M	
<a href="#">mexart-2018-10-25-chan-01.dat</a>	2018-10-25 19:05	33M	
<a href="#">mexart-2018-10-26-chan-00.dat</a>	2018-10-26 19:05	1.5K	
<a href="#">mexart-2018-10-26-chan-01.dat</a>	2018-10-26 19:05	1.5K	
<a href="#">mexart-2018-10-27-chan-00.dat</a>	2018-10-27 19:05	153K	
<a href="#">mexart-2018-10-27-chan-01.dat</a>	2018-10-27 19:05	146K	
<a href="#">mexart-2018-10-28-chan-00.dat</a>	2018-10-28 19:05	177K	
<a href="#">mexart-2018-10-28-chan-01.dat</a>	2018-10-28 19:05	169K	
<a href="#">mexart-2019-08-06-chan-01.dat</a>	2019-08-06 19:05	96M	

*Apache/2.4.10 (Debian) Server at www.rice.unam.mx Port 80*

Figura 2.9: Página Mexart

Cada día se debe realizar la verificación del día, si es el día correcto se va a realizar el respaldo, si ya termino el mes como se va a generar la

validación con los mismos datos del sistema deberá crearse la siguiente carpeta con el nuevo mes, en caso de que termine el año debe ser igual validar si ya terminó el año deberá generarse una nueva carpeta con el nuevo año.

### 2.5.1. Tolerancia a fallas

El sistema debe ser tolerante a dos tipos de fallas, las cuales son:

*Fallos permanentes o transitorios en los componentes hardware.* Como podría ser el caso de que alguno de los discos duros falle al momento de que se esté generando el respaldo, o cuando se requiera de una recuperación de información.

*Fallos permanentes o transitorios en los subsistemas de comunicación.* En este caso se podría dar en el momento que se queden sin comunicación tanto el cliente como el servidor de tal forma que no se podría generar el respaldo de ese día.

### 2.5.2. Recuperación de datos

Actualmente, la Universidad Autónoma de la Ciudad de México (UACM) cuenta con dos redes de Internet, la primera es una **red básica**, está red no es simétrica por lo tanto sus megabytes de subida de archivos y los megabytes de bajada no son iguales.

A diferencia de la segunda **red niba** que tiene la universidad, está red está dedicada a ciertos proyectos los cuales requieren más megabytes y que sus datos estén más protegidos y con una garantía de un ancho de banda dedicado únicamente a realizar el respaldo de la información sin que este más gente utilizando esta red.

La red que se utilizará en el desarrollo de este proyecto es la red normal debido que para utilizar la red niba debe ser llevado el servidor al SAIT de la Universidad para que se conecte mediante el cable de fibra Óptica.

Después se va a restringir al usuario root para que no pueda acceder a toda la información remotamente.

Se cambiará el puerto por defecto que es 22 al puerto que tenga disponible en este caso será 2244, como este sistema estará disponible todo el tiempo, se requiere que al momento de que inicie el servidor de forma automática entonces se active el OpenSSH. Creación de usuario mediante SSH.

Para recuperar información del servidor se debe generar una llave segura entre la máquina remota y servidor de respaldos, para esto se establece la conexión encriptada `ssh nombre@direccionip`, se genera la llave cuando pregunta por primera ocaón esta debe ser aceptada para continuar con el proceso de acceso.

# Capítulo 3

## Implementación del sistema de respaldos

### 3.1. Configuración del hardware del servidor

Debido a las características especiales del servidor es necesario realizar una instalación particular para un servidor que respaldará y manejará volúmenes grandes de datos. A continuación, se describen los pasos a seguir con la distribución elegida de Linux (CentOS 7).

Antes de iniciar la configuración se realizó la configuración del sistema RAID [8] desde la Basic Input/Output System; en español Sistema Básico de Entrada y Salida (BIOS) del Servidor3.1 pulsando la tecla Supr al entrar a la bios se va a la pestaña *Bios UEFI*, de ahí mover en las pestañas hasta llegar a la de configuración de los discos, en SATA TYPE se elige RAID, y se selecciona RAID5, debido a que se cuenta con 6 discos duros, y para el Raid 5 se requiere 3 o mas discos, esto con la finalidad de que en un futuro se termine el servidor con todos los discos que caben que es un total de 36 discos duros.

Primero se necesita dar formato a la memoria tipo flash/USB de mínimo 8 Gb para que se pueda grabar una imagen ISO de forma correcta del sistema operativo CentOS 7, de cualquier sistema operativo al ingresar la memoria en el caso de Windows cualquier versión dando clic derecho, luego se le da formatear, después formato rápido y sistema



Figura 3.1: Servidor Huitzilopochtli UACM

de archivos [9] FAT32 , en el caso de usar alguna distribución Linux de igual forma clic derecho y formatear. En la siguiente dirección se baja la ISO del Sistema Operativo <https://www.centos.org/download/> [10]. Se utilizará el programa unebootin, una vez generada la USB instalable (bootable) se comenzará con la instalación.

Al iniciar la instalación primero se escoge el idioma que en este caso

es español (México), después se elige la tarjeta de red correspondiente, la entrada para configurar su dirección IP, así como el Servidor DNS, cabe mencionar que estas direcciones fueron proporcionados acorde a las IPs que están disponibles en el plantel.

Se elige el entorno Base, que va a ser un Escritorio GNOME, donde se seleccionan todos los complementos para este entorno. Se asigna un nombre al servidor que es Huitzilopochtli asociado a su dirección IP.

Para realizar la configuración de las particiones que tendrá el servidor de respaldos, se generaran diversas particiones como son:

- Particiones para Datos

/home. Esta partición es donde se va a almacenar toda la información de todos los usuarios, como es archivos caché, fotos, archivos de libre office, etc, la recomendación de Red Hay es de 100 Gigabyte (Gb) como mínimo.

/data. Esta partición es necesaria cuando se va a almacenar información con características muy específicas como un sistema de archivos, donde se almacenarán archivos de gran tamaño como los obtenidos de un respaldo remoto.

- Particiones para el sistema

/usr. Guarda gran contenido del software, se necesita al menos 4 Gb de espacio, el proceso de arranque es mucho más complejo debido a que /usr contiene componentes de arranque críticos.

/boot. Es donde está el núcleo del sistema operativo, con otros archivos que se utilizan durante el arranque en esta partición con unos 250 Megabytes (Mb) es más que suficiente para la mayoría de usuarios. Es necesario realizar esta partición de /boot. cuando se desea instalar un sistema RAID [11]

/boot/efi. Los sistemas con UEFI deben tener un tamaño entre 50 y 150 Mb, teniendo su propia partición ya que necesitan un sistema de archivos en formato fat32.

/ La partición raíz es donde se instala el sistema operativo, aquí el mínimo recomendado por Red HAt es 3 Gb.

/swap. Es la memoria de intercambio esto quiere decir que se utiliza cuando no tienen suficiente memoria RAM, como mínimo debe tener 256 Mb.

Todas y cada una de las extensiones estarán en formato Ext4 [12] Ver figura 3.2. para revisar las particiones de la configuración particular del servidor.

El sistema de archivos formato Ext4, se creó hace unos años a raíz de que el sistema de archivos Format Ext 3 llenó su máxima capacidad que estaba limitado a 16 Terabytes (tb) en la arquitectura x86, a consecuencia de esto, se generó el formato Ext 64 donde este tiene un máximo de sistema de archivos de 232 bloques ( 16 tb donde cada bloque tiene un tamaño de 4kb) a 248 bloques.

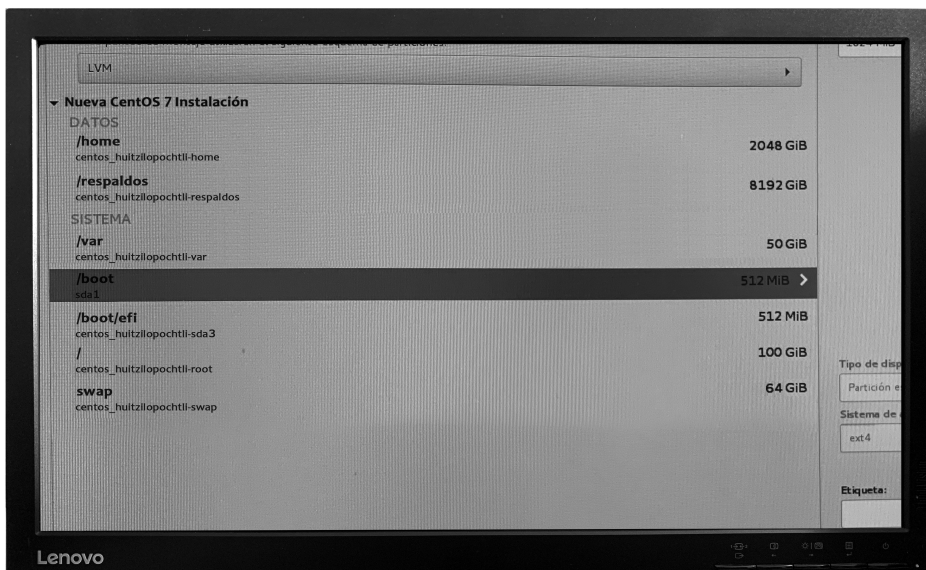


Figura 3.2: Particiones de Servidor Huitzilopochtli

Ya que se tienen las particiones se realizará la configuración de la

fecha que es Ciudad de México, Zona horaria, así como las contraseñas de root

Cuando se tiene la configuración correcta de particiones, versión del SO y herramientas para el servidor, se inicia toda la instalación del SO en este hardware.



Figura 3.3: Configuración de Sistema Operativo CentOS7

## 3.2. Configuración del software del sistema

Para la configuración del respaldo automático se generó un script el cual va a ser ejecutado mediante *Crontab*, es un proceso especial (daemon) de linux que se encarga de ejecutar tareas periódicamente, toda tarea que desee ejecutar se puede añadir en la siguiente ruta `etc/crontab`.

Básicamente es un archivo de texto, este archivo tiene contenido especial que estará formado por diversas instrucciones, como puede ser una simple línea con un instrucción como es un `ls` que se ejecute cada cierto tiempo o instrucciones más complejas como una lista con varios

scripts a ejecutar.

Este respaldo se va a generar cada 24 horas, pero no se respaldará el día actual, sino que se respaldará el día anterior, esto con la finalidad de que ya se haya terminado de guardar la información completa del día anterior.

Para habilitar el cron se tienen que proporcionar 5 datos que serán indicados con asteriscos \* \* \* \* \* en caso de que se requiera que se ejecute el script en todo momento.

- \* m: minuto
- \* h: hora
- \* dom: día de la semana
- \* mon: mes
- \* dow: día del mes

Se llama crontab desde la consola del servidor con la instrucción

```
crontab -e
```

Ya que abre la ventana del crontab se oprime la tecla i (comando para insertar), debido a que es una ventana de vi, se ingresan las siguientes líneas.

```
50 23 * * * /home/huitzilopochtli/Documents/scriptBasic/  
callisto/respaldoAutomatico.sh
```

La línea anterior indica como es que se va a generar el respaldo en el cual se le va a especificar en que minuto se va a generar, esto será cada minuto 50 de cada 23 horas donde se realizara todos los días de la semana, de todos los meses y todos los días del mes y de todo el año, ya que se especifica esta información se procede a poner la ruta del archivo llamado **respaldoAutomatico.sh** para que se ejecute durante el tiempo.

Después se guarda el archivo con :w y ya queda activado para ejecutarse justo al tiempo que se le especificó. Este archivo lo que contiene

es el script que hace que el respaldo se realice automáticamente.

El contenido del archivo **respaldoAutomatico.sh** es:

```
#!/bin/bash

dia=$(date +%d)
mes=$(date +%m)
anio=$(date +%Y)
hora=$(date +%H)
minuto=$(date +%M)
segundo=$(date +%S)

cd /home/Huitziloposhtli/Documents/Tesis/Respaldo

dia2=$(date --date "-1 days" +%d)

echo $dia2

wget -np -r http://www.rice.unam.mx/callisto/
lightcurve/$anio/$mes/$dia2/ >> errorCallisto.txt

wget -np -r http://www.rice.unam.mx/mexart/
ch00/$anio/$mes/$dia2/ >> errorMexart.txt
```

Este archivo inicia con `#!/bin/bash` Esos dos primeros caracteres tienen un nombre popular, shebang y permiten determinar qué intérprete se encargará de ejecutar el resto del script, de la fecha del mismo sistema se va a separar por día, mes, año, cada uno en una variable.

Para realizar el respaldo del día anterior, se tiene que sacar el día, mes y año, respecto a un día antes, ya que se tienen esos datos se realiza el respaldo con **wget** seguido de una url que se va a respaldar. Por cada carpeta que va a respaldar como se maneja por año, mes y día, entonces en lugar de poner cada dato fijo, se mete cada una de las variables en el script para que en automático vaya cambiando conforme vayan pasando los días.

Para el caso de la segunda página a respaldar ya no es necesario generar otro archivo, sino que en este mismo se agregará la segunda página a respaldar que es la de **Mexart**, indicando también las variables correspondientes para que cambien en automático.

### 3.3. Recuperación de datos

Para la recuperación de datos se realizó la siguiente configuración.  
Instalación del servidor OpenSSH.

```
sudo yum install openssh-server
```

Configuración del servidor SSH Configuración de SSH

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
```

restricción de usuario root

```
PermitRootLogin no
```

Cambio de Puerto

```
Port 2244
```

Reiniciar el servidor SSH

```
sudo systemctl reload sshd
```

Activación de OpenSSH.

```
sudo chkconfig sshd on
```

crear Usuario SSH

```
root@huitzilopochtli:/etc/ssh
Archivo Editar Ver Buscar Terminal Ayuda
[root@huitzilopochtli ssh]# adduser servidormorelia
```

Figura 3.4: Crear usuario

```
root@huitzilopochtli:/etc/ssh
Archivo Editar Ver Buscar Terminal Ayuda
[root@huitzilopochtli ssh]# sudo password servidormorelia
```

Figura 3.5: Crear contraseña

crear contraseña

Configuración del Firewall

```
root@huitzilopochtli:/etc/ssh
Archivo Editar Ver Buscar Terminal Ayuda
[root@huitzilopochtli ssh]# sudo firewall-cmd --permanent --add-port=2244/tcp
```

Figura 3.6: Configuración firewall

Abrir el puerto que recién se cambió

```
sudo firewall-cmd --permanent --add-port=2244/tcp
```

reiniciar el firewall

```
sudo firewall-cmd --reload
```

Probar conexión SSH desde otra máquina, se podrá hacer la conexión al servidor donde solo se necesita el usuario y su contraseña.

```
ssh -p 2244 servidormorelia@187.188.100.185
```

## 3.4. Pruebas de respaldo de la información

Se generaron scripts para realizar pruebas de cuanto tiempo tarda en generarse las pruebas.

**-np** (no parent), con esta instrucción se indica que no va a seguir ningún otro enlace a directorios superiores, solo estará en el directorio actual y uno hacia abajo y es equivalente a poner **-r -l1** donde **l** nos indica que nivel es en el que va a estar.

**-r** nos dice que va a realizar recursividad en el directorio que se encuentra, es decir si hay más carpetas abajo va a entrar a cada una recursivamente hasta regresar a la raíz.

**-i** indica que existe un archivo .txt el cual contiene una lista de las url a las cuales va ingresar.

**-c** Continúa la descarga si se detiene desde el punto donde se quedó. El siguiente comando es por cada página que se desea respaldar.

```
wget -np -r http://www.rice.unam.mx/callisto/lightcurve/  
2018/03/01/ -o reporte1.txt
```

En el siguiente script se muestran diversos archivos con extensión .txt donde cada archivo contendrá el comando anterior por cada número de carpeta a respaldar.

```
wget -np -r -i 5Carpetas.txt -o reporte5c.txt  
wget -np -r -i 10Carpetas.txt -o reporte10c.txt  
wget -np -r -i 20Carpetas.txt -o reporte20c.txt  
wget -np -r -i 30Carpetas.txt -o reporte30c.txt
```

Las pruebas realizan el respaldo de diversas carpetas por día.

- Respaldo de 1 carpeta.
- Respaldo de 5 carpetas.

- Respaldo de 10 carpetas.
- Respaldo de 20 carpetas.
- Respaldo de 30 carpetas.

El código anterior es para el respaldo de Callisto y guarda la información de esta antena.

Una vez generado el respaldo de cada punto anteriormente mencionado se genera una tabla en la cual se observan los tiempos y los megabytes que se guardaron en el respaldo como métrica

En el caso de **Mexart** se generan sus pruebas correspondientes para ver cuanta información y que tiempos son lo que tarda en hacerse el respaldo.

Para las pruebas, se generan los mismos archivos .txt a diferencia que estos van a contener las correspondientes url de mexart

```
wget -np -r http://www.rice.unam.mx/mexart/ch00/2018/01/01/ -o reporte1.txt
```

```
wget -np -r -i 5Carpetas.txt -o reporte5c.txt  
wget -np -r -i 10Carpetas.txt -o reporte10c.txt  
wget -np -r -i 20Carpetas.txt -o reporte20c.txt  
wget -np -r -i 30Carpetas.txt -o reporte30c.txt
```

En este caso las pruebas, realizan el respaldo de diversas carpetas por día.

- Respaldo de 1 carpeta.
- Respaldo de 5 carpetas.

- Respaldo de 10 carpetas.
- Respaldo de 20 carpetas.
- Respaldo de 30 carpetas.

Ahora veremos como es el comportamiento para el caso de la página de Mexart, debido a que no almacenan el mismo tipo de información, pues cada uno tiene distintos aparatos de medición

### 3.5. Pruebas de recuperación de información

Se van a realizar dos pruebas en distintas máquinas, la primera es una máquina de 1 tb en disco duro con 8 Gb de RAM, la cual cuenta con sistema operativo Fedora 19 (es una distribución GNU/Linux), estas pruebas constan de la lectura y escritura de archivos.

Se genera un archivo con un script del Shell Bash, el primero va a un archivo con datos aleatorios, primero va a crear un archivo de 1 Mb, después un archivo de 10 Mb, otro de 100 Mb, etcétera, este archivo el mismo script lo va a escribir y a leer dando unos valores los cuales se grafican para posteriormente, hacer una comparación con el Servidor marca SUPERMICRO, el cual tiene 16 tb de almacenamiento, con posibles expansiones para mayor capacidad y 32 Gigabytes de RAM, el cual va a tener sistema operativo CentOS 7 y todas las herramientas de respaldo y seguridad usando software libre.

En el Script de abajo se filtra por bs, indica que tanto la lectura como la escritura de los archivos se realicen en bloques de 1 Megabyte, si se pone menos, sería más lento, pero más seguro, y con más se arriesga a que se pierda información en el proceso de copiado [13].

**input file** archivo de entrada, if=archivo.txt

**output file** archivo de salida, of=archivo.txt

**pv** es una instrucción muy reservada, la cual funciona como tubería de terminal que mide los datos que pasan

**count** numero de bloques

```
dd if=origen —pv—dd of=destin.txt
```

El tiempo que se está tardando en copiar de un archivo a otro cada cierta cantidad de bloques será guardado en un archivo txt, del cual se analizará generando unas graficas de estos mismos.

Script para generar archivo con datos aleatorios (random) es:

```
#!/bin/bash
mb=1
for i in {1..7};
do
mb=$((mb*10))
echo $mb
dd if=/dev/urandom of=file2.txt count=1M bs=$mb &>>
datos2.txt

ls -lh file2.txt >> datos2.txt
echo "##### count=1M "$mb " #####
#####" >> datos2.txt
done
```

En el siguiente script se genera la prueba generando archivos vacios, este scrip se ejecutó tanto en el servidor como en la maquina laptop.

```
#!/bin/bash
mb=1
for i in {1..7};
do
mb=$((mb*10))
echo $mb
dd if=/dev/zero of=file.txt count=1M bs=$mb &>>
datos.txt
```

```
ls -lh file.txt >> datos.txt
echo "##### count = 1M"$mb "MB #####
#####"
>> datos.txt
done
```

Pruebas en una máquina 1 mencionada anteriormente.

El archivo inicialmente solo tenía la palabra Hola uacm.

Se generaron 3 tipos de archivos, la primera prueba se realizó con count= 14 kb, el segundo con un count= 254 kb y el tercero con count=1M, el script está dentro de un for (se repite 7 veces), el cual va a ir incrementando el valor de  $bs^2$ , entonces los valores serán 1, 10, 100, 1000, 10000, 100000, 1000000.

# Capítulo 4

## Análisis

### 4.1. Rendimiento de lectura y escritura

El rendimiento de la lectura y escritura de archivos es muy importante ya que con ello se puede tener una idea más concreta de las tazas sostenidas cuando se realiza el respaldo en el servidor a diferencia de una máquina común (PC).

En las siguientes gráficas mostradas a continuación se pueden observar que las marcas *X* son para las pruebas hechas en una laptop con características antes mencionadas en el capítulo 3, y las marcas \* que se observan son los datos del Servidor que es donde se está almacenando la información.

El comportamiento de las gráficas muestra que la laptop al inicio es más rápida debido a que las máquinas que se utilizan normalmente hacen procesos muy pequeños y en la prueba comienza con archivos de datos pequeños, cuando se realizan las pruebas con 1 Mb, la laptop comienza a volverse más lenta pues ahora sus procesos son mucho más pesados y pueden saturarse sus canales internos de comunicación y la memoria RAM.

La diferencia del servidor con la laptop es que el servidor comienza muy lento pues casi no tiene procesos que realizar, pero cuando comienza a realizar las pruebas con 1 Mb y más grandes, este se vuelve mucho más eficiente de tal forma que sus tiempos mejoran y es más eficiente su desempeño.

Por lo tanto se puede observar en las gráficas que al inicio la laptop es más rápida pero a determinado tamaño de bloques, los gráficos de los puntos se cruzan pues ahora son más procesos que se están ejecutando y esto es más eficiente para el sistema de archivos del servidor.

La primera tabla que obtenemos es de count =14 kb:

bloque L/E	segundos laptop (l)	segundos servidor (s)
1	0.00028017	0.000492356
10	0.000264036	0.000473308
100	0.000267667	0.000431449
1000	0.000378277	0.000578691
10000	0.00113191	0.00181058
100000	0.085652	0.0141676
1000000	2.1176	0.330749

Cuadro 4.1: Archivo vacío 14 kb

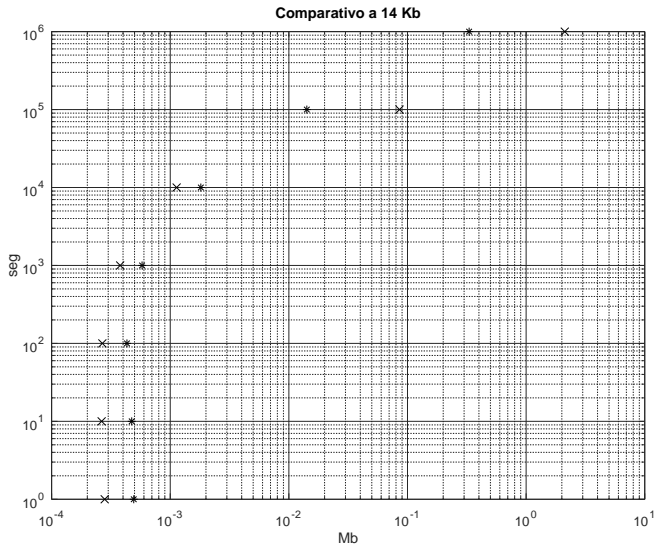


Figura 4.1: Archivo vacío 14 kb

Calculamos la pendiente para sacar el ancho de banda de la red de la universidad

bloque l/e	segundos l	segundos s
1	0.00168744	0.0020396
10	0.0015244	0.00148889
100	0.00174795	0.00165208
1000	0.00436332	0.00394697
10000	0.0237532	0.0273667
100000	0.757174	0.74075
1000000	9.77583	9.88837

Cuadro 4.2: Archivo vacío 254 kb

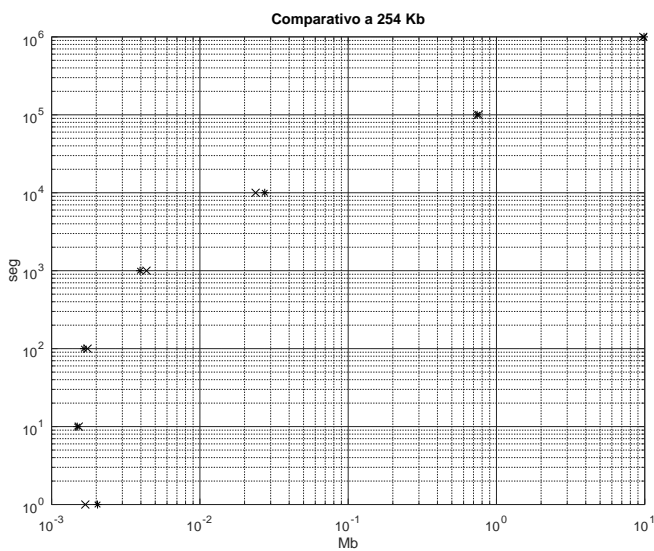


Figura 4.2: Archivo vacío 254 kb

bloque l/e	segundos l	segundos s
1	2.61661	2.23408
10	3.94073	2.47456
100	14.5328	2.79232
1000	127.645	22.2674
10000	223.706	155.423

Cuadro 4.3: Archivo vacío 1 Mb

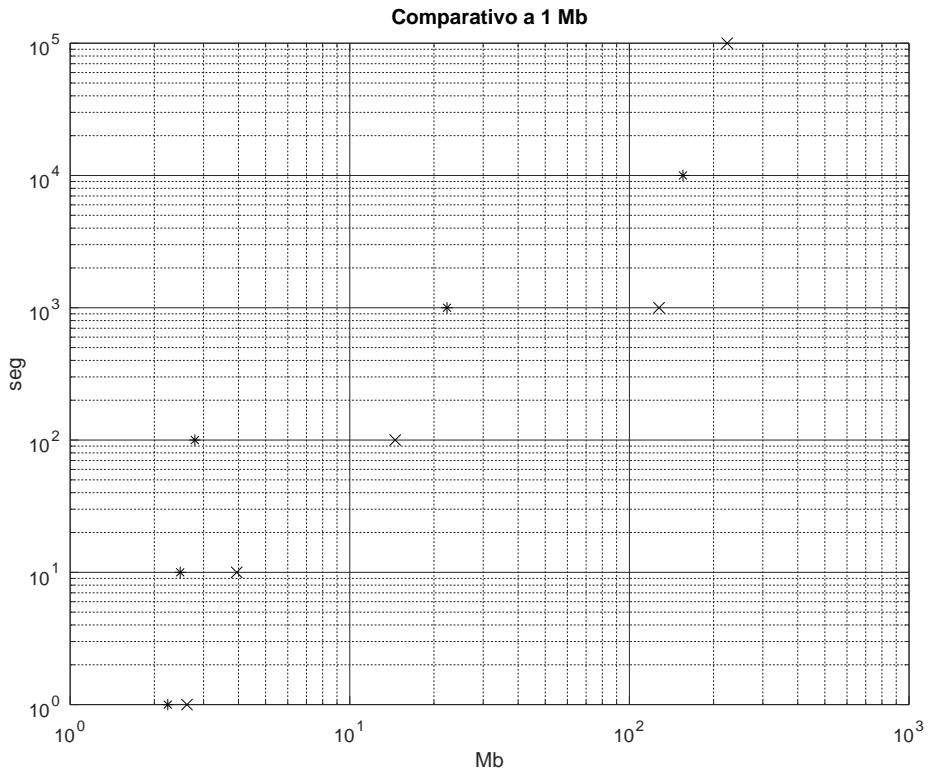


Figura 4.3: Archivo vacío 1 Mb

Se generaron otras tablas donde se puede observar cuantos Megabytes (Mb) por segundo se copian

bloque l/e	segundos l	segundos s
1	0.00043264	0.000661146
10	0.000579413	0.00217661
100	0.000467185	0.000641526
1000	0.00119654	0.00203677
10000	0.00815178	0.0155479
100000	0.110023	0.109981
1000000	2.23553	1.19782

Cuadro 4.4: Archivo random 14 kb

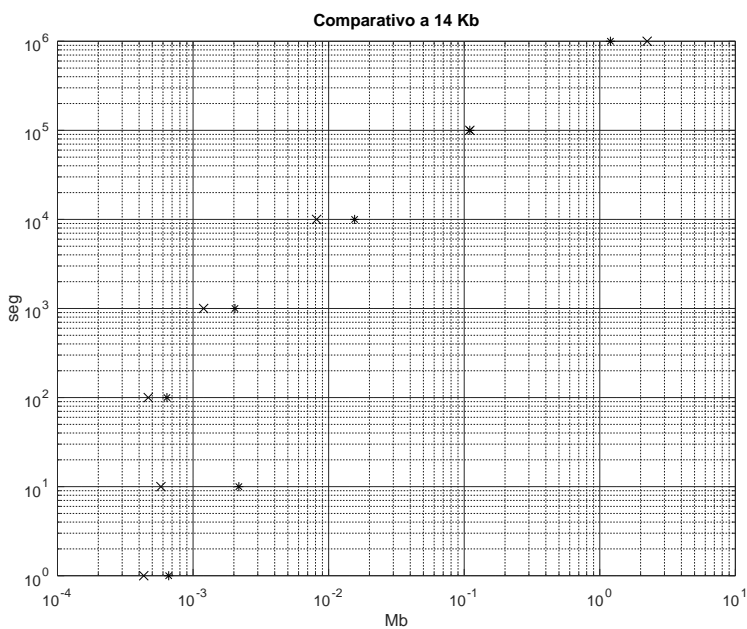


Figura 4.4: Archivo random 14 kb

bloque l/e	segundos l	segundos s
1	0.00140384	0.00176638
10	0.00155994	0.00148889
100	0.00388359	0.00694358
1000	0.0163344	0.0274637
10000	0.255487	0.18851
100000	4.2815	2.26962
1000000	43.6997	15.2704

Cuadro 4.5: Archivo random 254 kb

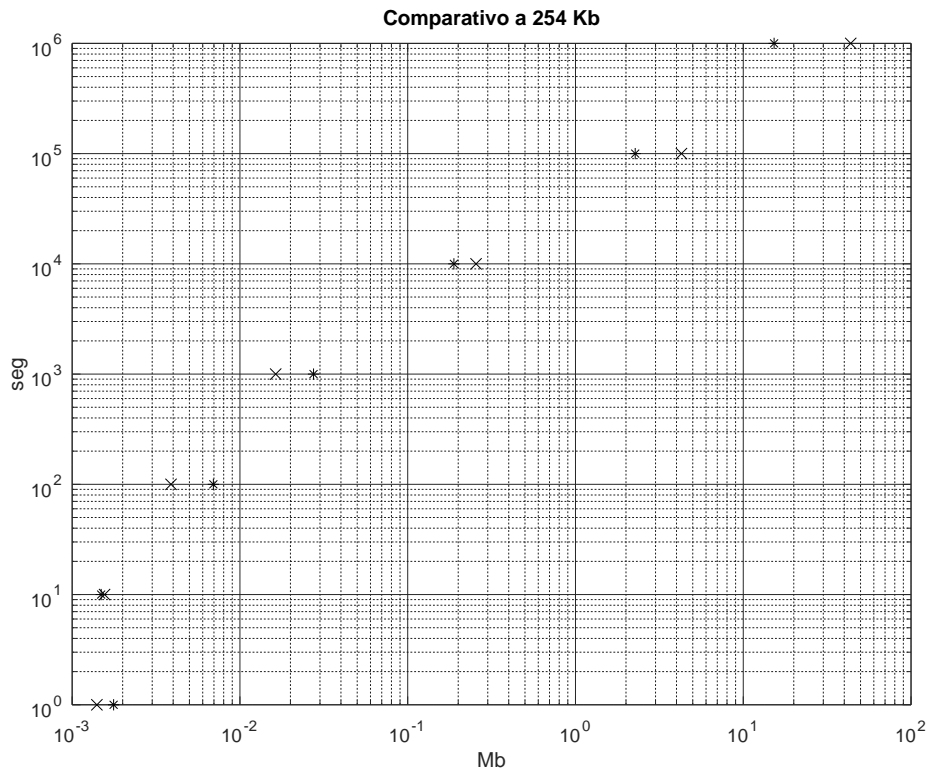


Figura 4.5: Archivo random 254 kb

bloque l/e	segundos l	segundos s
1	2.89783	2.76439
10	4.67415	3.49115
100	17.1288	9.19137
1000	142.618	67.5127
10000	231.709	266.268

Cuadro 4.6: Archivo random 1 Mb

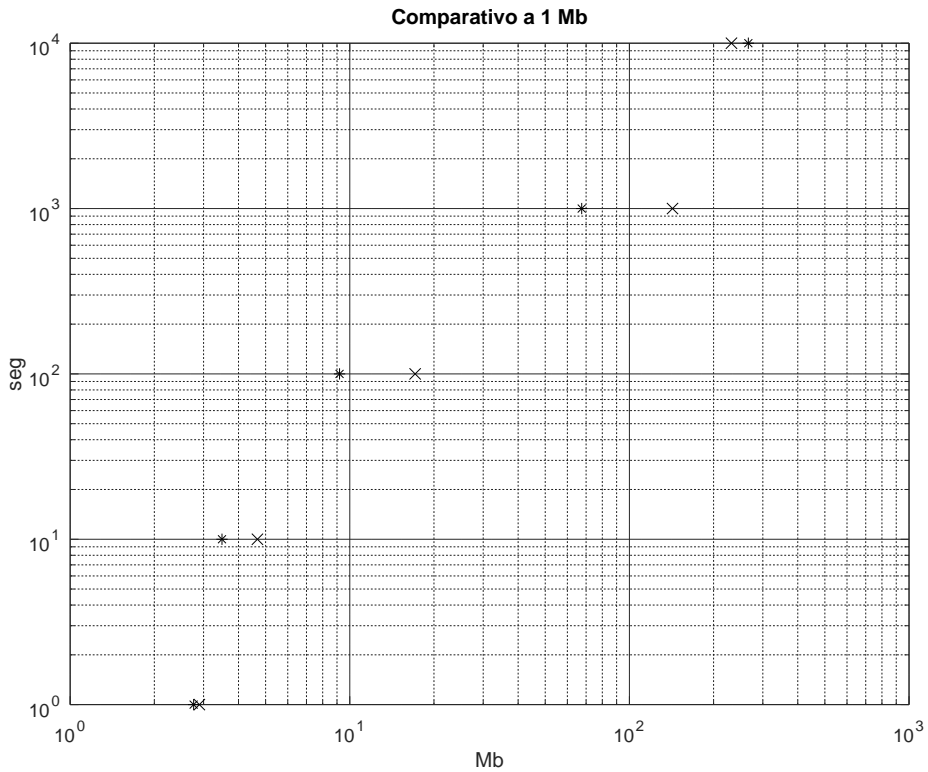


Figura 4.6: Archivo random 1 Mb

Estas son las gráficas del mismo script anterior, pero ahora se van a realizar las pruebas en el Servidor donde se está haciendo el respaldo.

## 4.2. Análisis de pruebas de respaldo

En las siguientes dos gráficas se verá el comparativo de las pruebas de respaldo, donde está la información del tiempo y el ancho de banda que se utilizó para hacer el respaldo, este respaldo se fue haciendo primero una carpeta, después 10 carpetas, 20 carpetas, 30 carpetas que son las correspondientes y representativas de un mes de respaldo.

N carpetas	seg Callisto	mb Callisto	seg Mexart	mb Mexar
1	17	43	36	43
5	60	211	56	209
10	93	421	186	608
20	147	792	573.6	1100
30	252	1210	1405.2	2800

Cuadro 4.7: Métrica Callisto megabyte/Segundos Transferencia

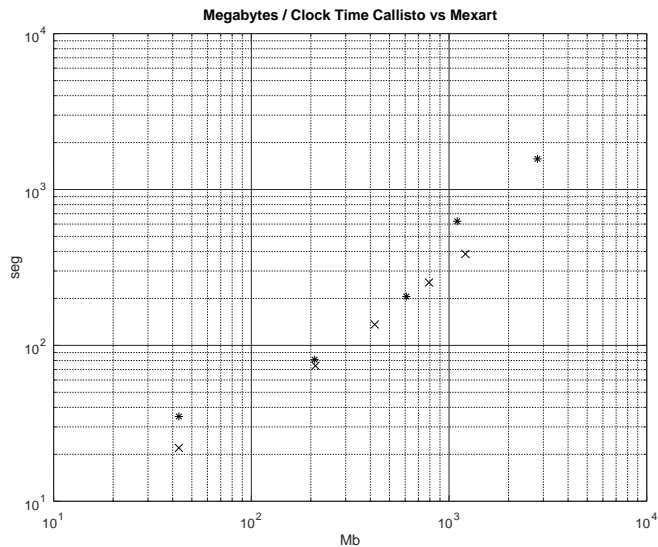


Figura 4.7: Callisto vs Mexart

$$T_f = 4.22 \quad T_i = 2.27$$

$$M_{bf} = 792 \quad M_{bi} = 421$$

$$\frac{792mb - 421mb}{253,2s - 136,2s} = 3,17mb/s \quad (4.1)$$

$$T_f = 252 \quad T_i = 147$$

$$M_{bf} = 1210 \quad M_{bi} = 792$$

$$\frac{1210mb - 792mb}{252s - 147s} = 3,98mb/s \quad (4.2)$$

En ambas gráficas se observa que los archivos que son respaldados son más pesados en Mexart, a pesar de que su información se guarda en el servidor principal cada 24 hrs, y la información que almacena en su mayoría son archivos .dat lo que significa que son archivos de datos crudos, guardan en directo del arreglo de antenas que reciben las señales.

N carpetas	clock time Callisto	mb Callisto	clock time Mexart	mb Mexar
1	22	43	35	43
5	74	211	81	209
10	136.2	421	205.8	608
20	253.2	792	352.2	1100
30	385.8	1210	1571.4	2800

Cuadro 4.8: Métrica Callisto vs Mexart megabyte/clock time Transferencia

$$T_f = 1571 \quad T_i = 625.2$$

$$M_{bf} = 2800 \quad M_{bi} = 1100$$

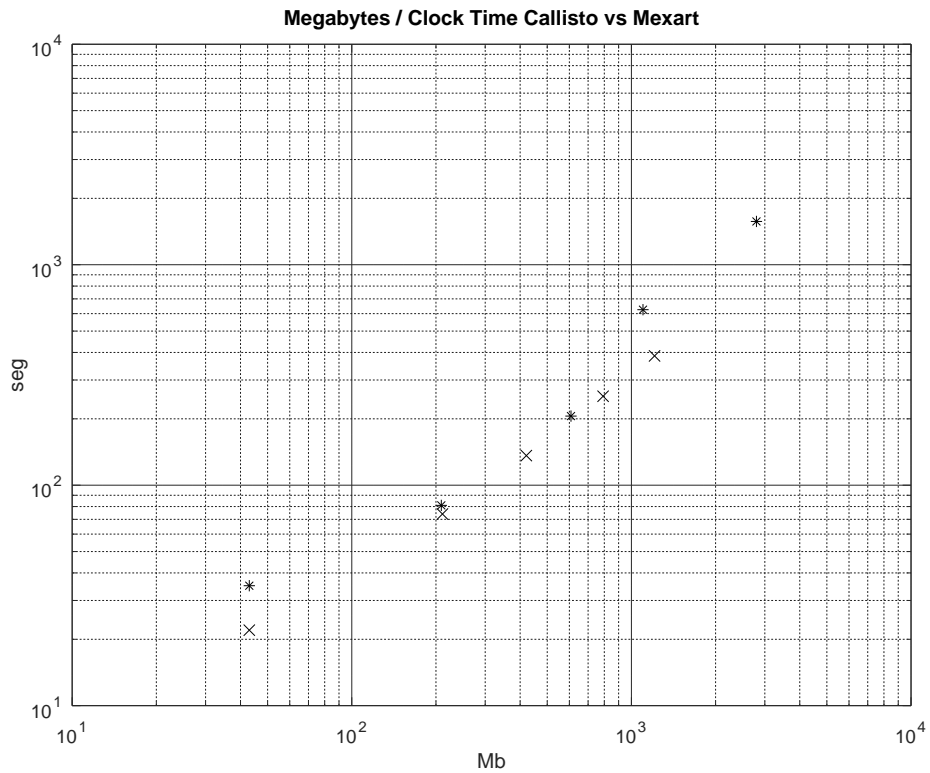


Figura 4.8: Callisto vs Mexart

$$\frac{2800mb - 1100mb}{1571,4s - 325,2s} = 1,62mb/s \quad (4.3)$$

$$T_f = 1405.2 \quad T_i = 573.6$$

$$M_bf = 2800 \quad M_{bi} = 1100$$

$$\frac{2800mb - 1100mb}{1405,2s - 573,6s} = 2,04mb/s \quad (4.4)$$

### 4.3. Proyectos futuros

El sistema desarrollado es el inicio de varias mejoras, debido a que solo se trabajó con el front-end esto es con la parte interna del sistema, este proyecto tendrá otras etapas como:

En un futuro se debe generar una Interface gráfica, esto con el objetivo de que realmente sea un respaldo de RICE y sus datos sean públicos de tal forma que estos mismos puedan ser visibles desde algún portal web.

También falta realizar minería de datos, la cual se puede trabajar de dos formas. La primera es analizar la información en bruto que está respaldada para ver si existe algún patrón que pueda afectar el clima espacial, cuando sucede algún evento de gran magnitud. La otra forma es realizar un análisis de las imágenes de la información que ya ha sido procesadas.

Otra forma de realizar un aporte con el proyecto es trabajar con la información que ya está procesada sobre las imágenes que de igual forma estaba almacenadas en el servidor, realizando un análisis de imágenes (segmentación, análisis de pixeles, etcétera.), para que de igual forma puedan buscar algún patrón que pueda ayudar a identificar eventos en curso que afecten la atmósfera o la actividad tecnológica



# Capítulo 5

## Conclusiones

Desarrollar este sistema es importante tanto para la Universidad Nacional Autónoma de México (UNAM) como para la Universidad Autónoma de la Ciudad de México (UACM) por que es el inicio de una colaboración interinstitucional que podría expandirse y convertirse en un proyecto de más alcance.

La ventaja de haberlo desarrollado con software libre es que se puede adaptar a las necesidades del cliente que en este caso es el Repositorio Institucional de Morelia Michoacán utilizando el lenguaje script bash para las diversas funciones del sistema (crontab para que se ejecute cada cierto tiempo el respaldo y se realice automáticamente con niveles de tolerancia).

En cuanto se tenga la necesidad de recuperar algún archivo, únicamente requiere de tener una conexión a Internet y tener una terminal en un sistema operativo usando software libre u otro, contar con el usuario y contraseña que se le ha generado y puede hacer la recuperación de datos en cualquier momento que se requiera.

Este sistema no termina aquí, solo es el comienzo de un gran proyecto donde en un futuro se debe generar su propia página web, para su consulta pública, generar minería de datos de las imágenes que están respaldadas donde se podrá buscar un patrón de las llamaradas solares que se registran.

Debido a que existen soluciones comerciales para el respaldo y recu-

peración de la información, estos por lo general resultan muy caros. Los desarrollos en ambientes de software libre y en universidades permiten entender y aprender como hacer las implementación y caracterizar con métricas básicas su desempeño aún en el caso de que puedan ser menos eficientes que las contrapartes comerciales.

Al desarrollar este sistema surgieron algunos obstáculos como es el realizar la solicitud de la red Niba con la que cuenta la universidad, a la cual no pude tener el acceso y era importante para el proyecto pues es una red dedicada, la cual podía ser una ventaja para este proyecto en el momento que decidan ampliarlo y no se realizaron estas pruebas para verificar la diferencia del ancho de banda entre ellas.

La solicitud de una ip pública fue difícil por cuestión de tiempo en lo que se realizó la solicitud a las autoridades correspondientes, otra dificultad es que la universidad no cuenta con un sistema de respaldo eléctrico fiable para tener el sistema activo sin que puedan existir fallas eléctricas.

El desarrollar el sistema mediante bash me dio ventajas como es que tiene fácil acceso al sistema, puede hacer interpretación con wget así como la planificación de actividades periódicas usando crontab, logrando la programación para la activación automática de los scripts cada cierto tiempo según sea necesario.

Otra ventaja de usar el sistema linux es que se puede utilizar las mismas herramientas que trae el sistema operativo como es utilizar la consola del sistema para crear los scripts sin necesidad de utilizar algún otro software, el descargar e instalarlo.

# Apéndice A

## Apéndice

### A.1. Creación de gráficas con octave

Para realizar el proceso de las gráficas, se utilizó octave, y tener la información que se desea graficar, pero con el se fueron generando diversos scripts

```
scpt01.m
```

```
mm = csvread('mtt.csv');  
loglog(mm(:,1),mm(:,2),'xk',mm(:,13),mm(:,14),'*k')  
xlabel("Mb")  
ylabel("seg")  
title("Comparativo a 14 Kb")  
grid on  
print -dpdf figc407v.pdf
```

```
scpt02.m
```

```
mm = csvread('mtt.csv');  
loglog(mm(:,3),mm(:,4),'xk',mm(:,15),mm(:,16),'*k')  
xlabel("Mb")  
ylabel("seg")  
title("Comparativo a 254 Kb")  
grid on  
print -dpdf figc408v.pdf
```

```
scpt03.m
```

```
mm = csvread('mtt.csv');
loglog(mm(:,5),mm(:,6),'xk',mm(:,17),mm(:,18),'*k')
xlabel("Mb")
ylabel("seg")
title("Comparativo a 1 Mb")
grid on
print -dpdf figc409v.pdf
```

scpt04.m

```
mm = csvread('mtt.csv');
loglog(mm(:,5),mm(:,6),'xk',mm(:,17),mm(:,18),'*k')
xlabel("Mb")
ylabel("seg")
title("Comparativo a 1 Mb")
grid on
print -dpdf figc409v.pdf
```

scpt05.m

```
mm = csvread('mtt.csv');
loglog(mm(:,5),mm(:,6),'xk',mm(:,17),mm(:,18),'*k')
xlabel("Mb")
ylabel("seg")
title("Comparativo a 1 Mb")
grid on
print -dpdf figc409v.pdf
```

scpt06.m

```
mm = csvread('mtt.csv');
loglog(mm(:,11),mm(:,12),'xk',mm(:,23),mm(:,24),'*k')
xlabel("Mb")
ylabel("seg")
title("Comparativo a 1 Mb")
grid on
print -dpdf figc412z.pdf
```

scpt07.m

```
mm = csvread('mtt.csv');
loglog(mm(:,26),mm(:,25),"xk",mm(:,30),mm(:,29),"*k")
xlabel("Mb")
ylabel("seg")
title("Megabytes / Segundos de Transferencia Callisto
vs Mexart")
grid on
print -dpdf figc413pr.pdf
```

scpt08.m

```
mm = csvread('mtt.csv');
loglog(mm(:,26),mm(:,25),"xk",mm(:,30),mm(:,29),"*k")
xlabel("Mb")
ylabel("seg")
title("Megabytes / Segundos de Transferencia Callisto
vs Mexart")
grid on
print -dpdf figc413pr.pdf
```



# Bibliografía

- [1] *Pro Data, Backup and recovery*, author=Steven Nelson, publisher=Apress Copyright © 2011 by Steven Nelson.
- [2] Francisco de Asís López Fuentes. *Sistemas Distribuidos*, chapter Capítulo 3: Modelos de arquitecturas. 2015 UNIVERSIDAD AUTÓNOMA METROPOLITANA, 2015.
- [3] Umesh Hodeghatta Rao and UmeshaNayak. *The InfoSec Handbook an introduction to information Security*. Apress open.
- [4] Derechos Reservados LANCE/SCiESMEX Instituto de Geofísica UNAM (2018). Servicio de clima espacial en México. url <http://www.sciesmex.unam.mx/>, 2018. Accedido 09-09-2019.
- [5] Copyright © 2018 Universidad Nacional Autónoma de México-UNAM. Repositorio institucional de clima espacial rice. url <http://www.rice.unam.mx/>, 2018. Accedido 25-10-2018.
- [6] Copyright © 2018 Universidad Nacional Autónoma de México-UNAM Instituto de Geofísica. Servicio de clima espacial en México. url <http://www.rice.unam.mx/instrumentacion/radio-telescopio-callisto/>, 2018. Accedido 22-07-2019.
- [7] MEXART 2019Design: HTML5 UP. Observatorio de centelleo interplanetario de coeneo michoacan, universidad nacional autonoma de México. url <http://www.mexart.unam.mx/>, 2018. Accedido 23-07-2019.
- [8] traducido por Juan Piernas Cánovas Jakob Østergaard. Raid-software. url <http://es.tldp.org/COMO-INSFLUG/COMOs/Software-RAID-Como/Software-RAID-Como.htmltoc4>, 1999. Accedido 20-02-2019.

- [9] Martin Farach-Colton Tokutek Rutgers Bradley C. Kuszmaul Tokutek MIT John Esmet Tokutek Rutgers, Michael A. Bender Tokutek Stony Brook. The tokufs streaming file system. url <https://www.usenix.org/system/files/conference/hotstorage12/hotstorage12-final52.pdf>, 1999. Accedido 19-04-2019.
- [10] © 2019 The CentOS Project. Centos. url <https://www.centos.org/download/>, 2019. Accedido 16-10-2018.
- [11] Garth Gibson David A Patterson and Randy H Katz. *A case for Redundant Arrays of Inexpensive Disks (RAID)*. Computer Science Division, Department of Electrical Engineering and Computer Sciences 571 Evans Hall, University California Berkeley, CA 94720.
- [12] Federico Bergero Erwin Meza Gunnar Wolf, Esteban Ruiz. *FUNDAMENTOS DE SISTEMAS OPERATIVOS*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO. INSTITUTO DE INVESTIGACIONES ECONÓMICAS. FACULTAD DE INGENIERÍA.
- [13] Arnold Robbins. *Bash Poker Reference*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2016.