

# Memoria de la Práctica

## ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS

### *DATSI*

*Vidal Peña, Arturo*

w140307

25 de junio de 2021

## Índice

<b>1</b>	<b>Introduccion</b>	<b>2</b>
1.1	Funciones auxiliares	2
1.1.1	get_config_lines	2
1.1.2	print_result	2
1.1.3	check_and_install_software	2
1.1.4	check_if_host_is_known	2
1.1.5	get_lvm_names_and_sizes	2
1.2	Programa principal	3
<b>2</b>	<b>Dificultades encontradas</b>	<b>4</b>
2.1	Dificultades personales	4
2.2	Mejoras a considerar sobre el proyecto	4
<b>3</b>	<b>Configuración <i>Mount</i></b>	<b>5</b>
<b>4</b>	<b>Configuración RAID</b>	<b>6</b>
<b>5</b>	<b>Configuración LVM</b>	<b>7</b>
<b>6</b>	<b>Configuración NIS</b>	<b>8</b>
6.1	Servidor NIS	8
6.2	Cliente NIS	8
<b>7</b>	<b>Configuración NFS</b>	<b>10</b>
7.1	Servidor NFS	10
7.2	Cliente NFS	10
<b>8</b>	<b>Timeline del proyecto</b>	<b>12</b>
	<b>Bibliografía</b>	<b>16</b>

# 1. Introduccion

En este documento se detalla el proceso de realización del proyecto asignado. Dicho proyecto consiste en la realización de un script[1] en lenguaje Bash que sea capaz de instalar y configurar de forma autónoma un conjunto de servicios en un grupo de máquinas interconectadas en una red local.

## 1.1. Funciones auxiliares

Aquí se encuentran definidas funciones auxiliares a los métodos principales del proyecto.

### 1.1.1. `get_config_lines`

Este método lee el fichero de configuración pasado como argumento y extrae cada una de sus líneas en una variable, de nombre `lines`, usada como array.

### 1.1.2. `print_result`

Recibe un valor numérico de retorno y la función desde la que se llamó al comando que devolvió dicho valor. Imprime un mensaje informando de éxito, o especificando la función en la que se dio error.

### 1.1.3. `check_and_install_software`

Este método recibe la dirección IP local y la de destino de la configuración, junto a un listado de los paquetes necesarios para ejecutar la función llamante.

Este método comprueba si ambas direcciones IP coinciden, en cuyo caso la instalación será local. En otro caso será sobre una máquina en remoto, para lo que necesitará instalar localmente (si no está previamente instalado) el paquete `sshpass`, para poder operar de manera cómoda en la máquina remota.

Después, comprobará si el paquete ya se encuentra instalado en la máquina objetivo, e instalará aquellos que no lo estén.

### 1.1.4. `check_if_host_is_known`

Este método comprueba si la IP pasada como argumento existe en el fichero `~/.ssh/known_hosts`. En caso de no existir, añade una nueva clave ssh para dicha IP.

### 1.1.5. `get_lvm_names_and_sizes`

Esta función solamente es llamada desde el método principal LVM.

Recibe una lista ordenada de los nombres y tamaños de los distintos volúmenes lógicos a crear, obtenidos del fichero de configuración.

Comprueba los distintos pares `<Nombre, Tamaño>` que encuentra y los separa de manera ordenada en dos listas diferentes, una para nombres y otra para tamaños, estando todos los pares en los mismos índices de ambas listas.

En el caso de que alguno no esté escrito adecuadamente, finaliza la ejecución con un error.

## 1.2. Programa principal

Lo primero que comprueba el programa principal es si usamos el número de argumentos correcto (uno, el fichero de configuración del cluster); o si usamos los argumentos de ayuda `-h`, `--help`, en cuyo caso nos imprime un mensaje de uso.

También comprueba si el fichero de configuración especificado existe y es un fichero (no un directorio, por ejemplo); y si estamos ejecutando el script como superusuario.

Una vez hechas todas las comprobaciones, lee el fichero de configuración y extrae la información del mismo (IP de destino, mandato y fichero de configuración para el mandato). Todo ello, siempre que cumpla con las normas de sintaxis del fichero, para lo que contrasta la línea contra una serie de expresiones regulares (RegEx), que filtrarán las líneas comentadas y avisará si alguna línea está mal escrita.

Si cumple las RegEx, llamará al servicio que se requiera para cada línea.

## **2. Dificultades encontradas**

### **2.1. Dificultades personales**

Una de las primeras dificultades con las que nos encontramos fue la falta de tiempo que todos los miembros del equipo teníamos, debido en gran parte a estar compaginando estudios y trabajo. A pesar de ello, nos comprometimos en aportar todo lo posible para sacar adelante el proyecto.

Sin embargo, durante el transcurso del semestre, ninguno de los tres miembros aportamos lo suficiente para presentarla a tiempo. Aún así, siendo el que más aportó durante ese tiempo, decidí disolver el equipo y presentar el proyecto en solitario en la convocatoria extraordinaria.

Debido a una mala organización, falta de tiempo y recursos, no ha sido posible implementar el servicio de Backup, tanto para la parte cliente como servidor.

Por el mismo motivo, tampoco ha sido posible solucionar todos los errores que la implementación actual conllevan.

Esto sugiere posibilidades de mejora en la planificación del trabajo y del mantenimiento del mismo, junto a la posibilidad de ampliar las opciones de uso del proyecto ya implementado.

### **2.2. Mejoras a considerar sobre el proyecto**

Considero que este es un proyecto como muy pocos son en la carrera, que requiere tanto de conocimientos básicos como de gusto por la administración de sistemas.

Sin embargo, le veo un punto de mejora: sería interesante, dado que el principal uso de las máquinas virtuales para las pruebas ha sido mediante la terminal, contar con una imagen de Docker con la que disponer en contenedores interconectados del entorno necesario para el proyecto.

Considero que sería una mejora útil para aquellos estudiantes con pocos recursos en su máquina personal con los que mantener levantadas varias máquinas virtuales simultáneamente, con el objetivo único de probar la implementación.

### 3. Configuración *Mount*

Esta función se encarga de montar el Sistema de Ficheros dado en el directorio de destino especificado, creándolo en caso de que no existiera.

Para ello, lo primero que hace es usar el método auxiliar `get_config_lines` para obtener las líneas del fichero de configuración, en las que aparece el nombre del sistema de ficheros a montar y del directorio de destino de montaje. Las líneas deberán ser de la siguiente manera:

1. Nombre del dispositivo
2. Directorio de montaje

Luego, hace uso de la función `check_and_install_software` para instalar, sea local o remotamente, los paquetes necesarios para realizar el montaje. Estos paquetes son:

- **Montaje local:**

- `mount`

- **Montaje remoto:**

- `sshpass` localmente
- `mount` en el equipo de destino

Finalmente, crea el directorio de destino si no existía y luego monta el sistema de ficheros en él, haciendo uso del comando `mount[2]`.

## 4. Configuración RAID

Esta función creará un sistema RAID en la máquina de destino especificada. Las líneas del fichero de configuración del servicio se definen como sigue:

1. Nombre del dispositivo RAID
2. Nivel/Tipo de RAID
3. Dispositivos que lo componen

Después de obtener las líneas pertinentes del fichero de configuración, comprueba e instala los paquetes necesarios usando el método auxiliar `check_and_install_software`:

- Localmente:
  - `mdadm`
- En remoto:
  - `sshpass` localmente
  - `mdadm` en la máquina remota

Finalmente, usará el comando `mdadm[4]` en la máquina de destino (o mediante el comando `sshpass` en local) para crear el dispositivo RAID con el nombre, nivel y dispositivos especificados en el fichero de configuración.

## 5. Configuración LVM

Este método crea un Volumen Lógico en la máquina destino especificada.

Comprobará, a partir de la obtención de las líneas de configuración, que haya al menos tres líneas, puesto que el volumen lógico queda definido de la siguiente forma:

1. El nombre del volumen
2. Lista de dispositivos que lo componen
3. Nombre del volumen 1 y tamaño del mismo
4. Nombre del volumen 2 y tamaño del mismo
5. Nombre del volumen 3 y tamaño del mismo
6. etc.

En caso de no haber al menos tres líneas en el fichero de configuración, devolverá error.

Seguidamente, al igual que en los servicios anteriores, instalará el software necesario en las máquinas:

- Destino local:
  - pvcreate
  - vgcreate
  - lvcreate
- Destino remoto:
  - sshpass localmente
  - pvcreate en la máquina remota
  - vgcreate en la máquina remota
  - lvcreate en la máquina remota

También obtendrá, a partir de la tercera línea del fichero de configuración (y de las siguientes, si procediera), una lista con todos los nombres de los dispositivos y otra con todos los tamaños. Ambas listas están ordenadas de forma que compartan el índice, según la función auxiliar `get_lvm_names_and_sizes` definida anteriormente.

Finalmente, seguirá la siguiente secuencia:

1. Proceder a crear un grupo con la lista de dispositivos, con el comando `pvcreate[3]`.
2. Crear el volumen lógico asociado a ese grupo, con el nombre especificado, usando el comando `vgcreate[3]`.
3. Añadir los dispositivos y tamaños especificados, en orden, con el comando `lvcreate[3]`.

## 6. Configuración NIS

Esta función, al igual que las siguientes, se compone de una configuración distinta para máquinas servidor y cliente.

### 6.1. Servidor NIS

Después de obtener la línea del fichero de configuración, la cual contiene únicamente el nombre del que será el servidor NIS, instala los siguientes paquetes software<sup>1</sup>:

- Destino local:
  - nis
- Destino remoto:
  - sshpass localmente
  - nis en la máquina remota

Después de esto, configura los siguientes ficheros creados por el programa de instalación[7]:

- `/etc/default/nis`: Al que modificará la variable **NISSERVER**, dándole el valor *master*.
- `/etc/ypserv.securenets`: Comentará la dirección IP por defecto (0.0.0.0) y añadirá la de la red a la que está conectada la máquina, con máscara /24 (255.255.255.0).
- `/etc/default/nis`: Modificará también la variable **MERGEGROUP**, de *FALSE* a *TRUE*.
- `/etc/hosts`: Añadirá la dirección IP de red, junto al nombre del host y el nombre del servidor NIS.

Para terminar, aplica la configuración con el mandato `/usr/lib/yp/ypinit -m` y reinicia el servicio NIS creado.

### 6.2. Cliente NIS

Una vez obtenidas las líneas del fichero de configuración, que contiene el nombre del servidor NIS (como en la configuración de la parte servidor) y la dirección IP de dicho servidor; instala los siguientes paquetes:

- Destino local:
  - nis
- Destino remoto:
  - sshpass localmente

---

<sup>1</sup>**NOTA:** Al realizar la instalación, es requisito que el usuario introduzca el nombre del servidor NIS.



- nis en la máquina destino

Luego, configura los siguientes ficheros creados al instalar el software[6]:

- `/etc/yp.conf`: Añade el nombre del servidor y su dirección IP.
- `/etc/nsswitch.conf`: Añade la opción "nis.<sup>en</sup> los grupos `passwd`, `group`, `shadow` y `hosts`.
- `/etc/pam.d/common-session`: Añade la configuración para crear el directorio HOME al inicio de la sesión automáticamente.

Luego, añade el nuevo cliente al servidor, para poder usarse en el próximo inicio.

Para terminar, pregunta al usuario si quiere reiniciar el sistema, para poder empezar usar el servidor NIS ya configurado. En función de la respuesta (si no la hay, asumirá una respuesta afirmativa) reiniciará el sistema.

## 7. Configuración NFS

### 7.1. Servidor NFS

Primeramente, recoge las líneas del fichero de configuración, las cuales definen las rutas de los distintos directorios a exportar.

Después, instala el software necesario en la máquina:

- Destino local
  - nfs-kernel-server
- Destino remoto
  - sshpass en la máquina local
  - nfs-kernel-server en la máquina remota

Una vez instalado, modificará la variable **Domain** con el nombre de dominio de la máquina en el fichero `/etc/idmapd.conf`; y después, para cada uno de los directorios especificados en el fichero de configuración, añadirá los siguientes datos, en líneas por cada directorio a exportar[5]:

- Ruta del directorio
- Dirección IP de la red en la que se comparte, con máscara /24
- Las opciones para el mismo, siendo las siguientes:
  - rw: marca permiso de lectura y escritura
  - sync: habilita la sincronización del directorio
  - fsid=0: asigna el id 0 al nuevo Sistema de Ficheros
  - no\_root\_squash: habilita permisos de root
  - no\_subtree\_check: deshabilita la comprobación del subárbol de directorios

Por último, reiniciará el servicio para cargar la nueva configuración.

### 7.2. Cliente NFS

Primero obtiene las líneas del fichero de configuración, el cual contiene, por cada línea:

- La dirección IP del servidor NFS
- La ruta del directorio remoto
- El punto de montaje del nuevo directorio

Después, instala el software necesario:

- Destino local:
  - nfs-common
  - mount

- Destino remoto:
  - sshpass en la máquina local
  - nfs-common en la máquina remota
  - mount en la máquina remota

Para la configuración, primero modifica la variable **Domain** con el nombre de dominio de la máquina, en el fichero `/etc/ldapd.conf`, tras lo cual, reinicia el servicio **nfs-common** con la nueva configuración.

Luego, para cada línea del fichero de configuración del servicio, monta el directorio por nfs desde la ruta en la máquina de origen a la ruta en la máquina de destino.

Finalmente, añade la relación entre cada pareja de directorios al fichero `/etc/fstab`, de forma que se monte cada vez que se inicie sesión.

## 8. Timeline del proyecto

A continuación se lista el histórico de versiones del proyecto, de acuerdo con la historia extraída de la herramienta Git.

- **Fecha: vie, 2 de octubre, Autor: Arturo Vidal Peña**  
Create README.md
- **Fecha: sáb, 3 de octubre, Autor: Arturo Vidal Peña**  
Estructura de la memoria
- **Fecha: jue, 8 de octubre, Autor: Javier Perez Martin**  
delete all comments and empty lines from the file
- **Fecha: dom, 11 de octubre, Autor: Javier Perez Martin**  
echo bad format line
- **Fecha: jue, 15 de octubre, Autor: Arturo Vidal Peña**  
Enunciado. Archivos de configuracion vacios. Funciones en script para facilitar desarrollo
- **Fecha: jue, 22 de octubre, Autor: Arturo Vidal Peña**  
Refactor functions in auxiliar files
- **Fecha: jue, 19 de noviembre, Autor: Arturo Vidal Peña**  
Primer intento mount
- **Fecha: sáb, 21 de noviembre, Autor: Alfonso Sudara Padilla**  
lvm
- **Fecha: dom, 29 de noviembre, Autor: Arturo Vidal Peña**  
Mount funciona tanto en local como en host remoto. Añadido fichero de prueba.
- **Fecha: dom, 6 de junio, Autor: Arturo Vidal Peña**  
Finished RAID configuration & tests.
- **Fecha: dom, 6 de junio, Autor: Arturo Vidal Peña**  
Started lvm. Initiated w/ example test & config
- **Fecha: mar, 8 de junio, Autor: Arturo Vidal Peña**  
Completed lvm for both local & remote execution.
- **Fecha: mar, 8 de junio, Autor: Arturo Vidal Peña**  
Added documentation to common functions.

- Fecha: mar, 8 de junio, Autor: Arturo Vidal Peña  
Fixed typo. Added package installation call.
- Fecha: sáb, 19 de junio, Autor: Arturo Vidal Peña  
320tr First approach on NIS Server. Still fails.
- Fecha: dom, 20 de junio, Autor: Arturo Vidal Peña  
320tr Added config & test cases.
- Fecha: dom, 20 de junio, Autor: Arturo Vidal Peña  
320tr Added documentation URL.
- Fecha: dom, 20 de junio, Autor: Arturo Vidal Peña  
320tr Changed method name for NIS Server.
- Fecha: dom, 20 de junio, Autor: Arturo Vidal Peña  
320tz Updated gitignore to ommit LaTeX auxiliar files.
- Fecha: dom, 20 de junio, Autor: Arturo Vidal Peña  
320tz Removed partners from proyect memoir.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Added Bash script to automate git log and graph info recopilation & write it into LaTeX files.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Initial git timeline.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Initial git graph.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Initial difficulties statement.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Introduction, main program & common functions explained.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Added git-graph directory to gitignore as it's not needed.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña  
320tz Added includes dir with git-graph image & git log file.
- Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña

320tz Renamed functions LaTeX files.

- **Fecha: mar, 22 de junio, Autor: Arturo Vidal Peña**

320tz Updated main LaTeX files with current file names.

- **Fecha: mié, 23 de junio, Autor: Arturo Vidal Peña**

320tz Updated gitignore to ommit LaTeX auxiliar files & accept correct ones.

- **Fecha: mié, 23 de junio, Autor: Arturo Vidal Peña**

320tz Updated gitignore to ommit LaTeX auxiliar files & accept correct ones.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Updated gitignore to ommit LaTeX auxiliar files & accept correct ones.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Added LVM explanation.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Added introduction explanation.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Added mount explanation.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Added raid explanation.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Added main program explanation.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Added bibliography.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tz Removed git graph for clarity.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

Fixed mount & main program

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tq NIS client configuration implemented.

- **Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña**

320tu NFS Server implemented.

- Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña  
320tt NFS Client implemented.
- Fecha: jue, 24 de junio, Autor: Arturo Vidal Peña  
320tz Added NIS Server explanation.
- Fecha: vie, 25 de junio, Autor: Arturo Vidal Peña  
320tz Added NIS Client explanation.
- Fecha: vie, 25 de junio, Autor: Arturo Vidal Peña  
320tz Removed commits sha from output.
- Fecha: vie, 25 de junio, Autor: Arturo Vidal Peña  
320tz Added NFS Client explanation.
- Fecha: vie, 25 de junio, Autor: Arturo Vidal Peña  
320tz Final changes to memoir.

# Bibliografía

- [1] Gabriel Cebrián Márquez. *Administración de sistemas UNIX - Proyecto práctico. Script maestro para la configuración de un cluster Linux*.
- [2] ManPages. *Debian Manpages - mount(8) - mount*. URL: <https://manpages.debian.org/testing/mount/mount.8.en.html>. (accessed: 29.11.2020).
- [3] Debian Wiki. *Debian - Configure LVM*. URL: <https://wiki.debian.org/es/LVM>. (accessed: 08.06.2021).
- [4] Server World. *CentOS 7 - Configure RAID 1*. URL: [https://www.server-world.info/en/note?os=CentOS\\_7&p=raid1](https://www.server-world.info/en/note?os=CentOS_7&p=raid1). (accessed: 06.06.2021).
- [5] Server World. *Debian 7.0 Wheezy - Configure NFS Server*. URL: [https://www.server-world.info/en/note?os=Debian\\_7.0&p=nfs&f=1](https://www.server-world.info/en/note?os=Debian_7.0&p=nfs&f=1). (accessed: 20.06.2021).
- [6] Server World. *Debian 7.0 Wheezy - Configure NIS Client*. URL: [https://www.server-world.info/en/note?os=Debian\\_7.0&p=nis&f=2](https://www.server-world.info/en/note?os=Debian_7.0&p=nis&f=2). (accessed: 20.06.2021).
- [7] Server World. *Debian 7.0 Wheezy - Configure NIS Server*. URL: [https://www.server-world.info/en/note?os=Debian\\_7.0&p=nis&f=1](https://www.server-world.info/en/note?os=Debian_7.0&p=nis&f=1). (accessed: 20.06.2021).