

Memoria de “Práctica Script maestro para la configuración de un cluster Linux”

Integrantes:

Ayuso Fernandez Jorge, Bautista Salinero Jaime, Osornio Díaz Emiliano

1. Introducción

Nuestra práctica consta de un fichero principal, *configurar_cluster.sh*, que es el script encargado de analizar un fichero de configuración creado por el usuario, que contiene la información de los servicios que se van a configurar. Cada una de las líneas de dicho fichero está formada de tres parámetros: ip de la máquina en la que se va a configurar el servicio, servicio que se desea configurar y el archivo de configuración específico para dicho servicio.

Antes que nada verificamos que el usuario esté proporcionando un fichero de configuración. Después se analiza dicho fichero línea por línea, ignorando todas aquellas vacías o que comienzan con el carácter '#'. Ya que tenemos las líneas analizamos si estas tienen los tres parámetros necesarios para el despliegue de los servicios, se analizan dichos parámetros, se ejecutan y se copian los scripts correspondientes a cada servicio según sea el caso a la máquina destino realizando las conexiones a través de ssh.

Se separó cada uno de estos servicios en un script diferente para una mejor legibilidad en el código.

Complicaciones sobre esta parte fue una en específico y era que al ejecutar ciertos comandos, en nuestro caso ssh, nuestro programa rompía el ciclo que analizaba línea por línea el archivo de configuración y terminaba ejecutando un solo servicio, esto lo solucionamos agregando la bandera -n al hacer las conexiones con ssh.

2. Servicios

En esta sección explicaremos a detalle el funcionamiento y los problemas que nos encontramos en el desarrollo de cada uno de los servicios.

2.1 Mount

El programa en el que está desarrollado este servicio es mount.sh.

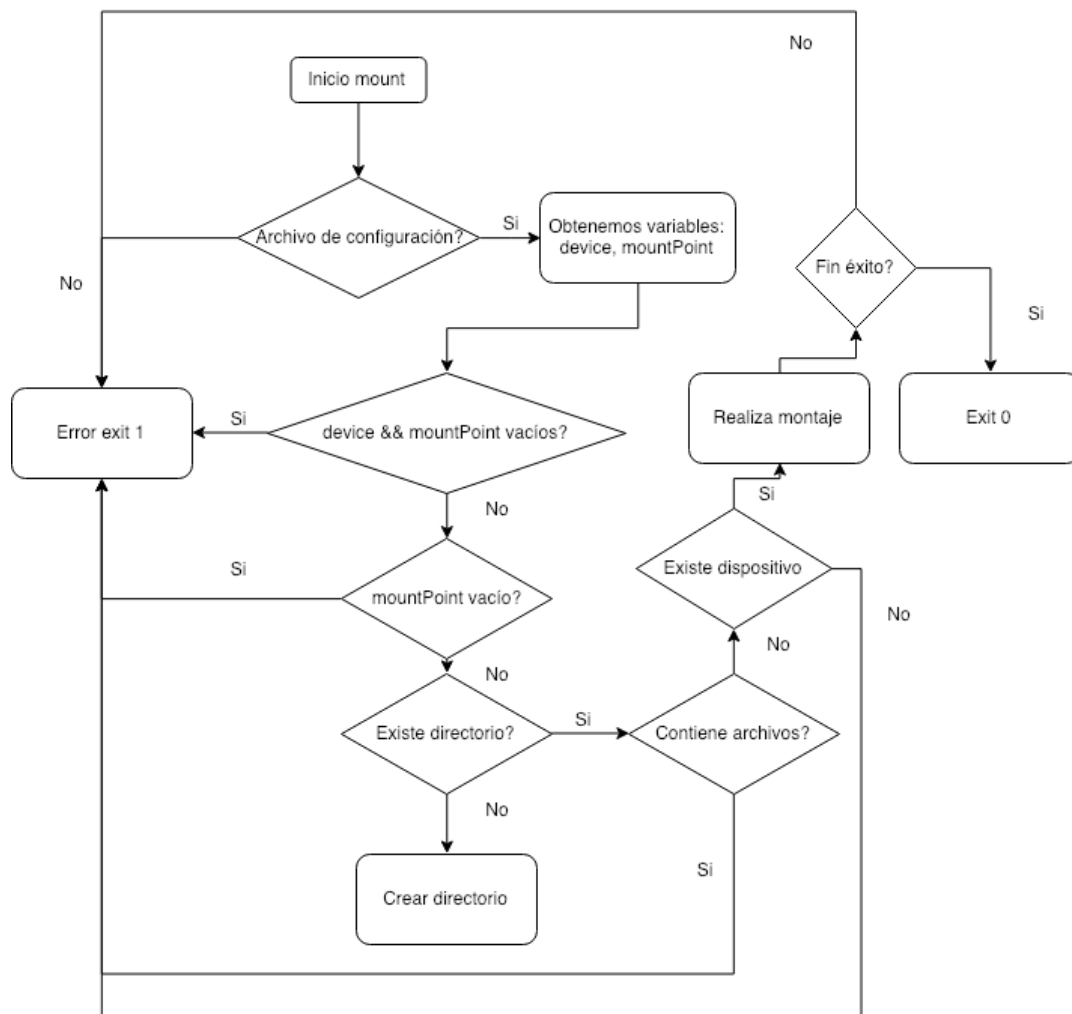
Como comprobaciones generales a lo largo del script se realiza:

- Que se haya especificado un archivo de configuración.
- Que se cuente con los dos parámetros necesarios para realizar el montaje. (dispositivo y punto de montaje)
- Que el directorio esté creado, sino se crea.

- Que el directorio no contenga ficheros.
- Que el dispositivo exista.

Una vez que se realizan todas estas verificaciones, obtenemos el tipo de dispositivo que se quiere montar y finalmente se realiza el montaje, modificando a su vez el fichero /etc/fstab para que estos cambios se mantengan al reiniciar la máquina.

A continuación se muestra un diagrama que resume el funcionamiento de dicho script:



Para comprobar su funcionamiento utilizamos la herramienta gparted, con esta creamos distintas particiones en diferentes discos duros creados sobre la máquina virtual, realizamos su formateo y creación de sistema de ficheros y una vez realizados estos pasos, verificamos que al realizar la función de mount desde configurar_cluster.sh con el respectivo archivo de configuración lograra hacer el montaje con éxito.

2.2 Raid

El programa en el que está desarrollado este servicio es raid.sh

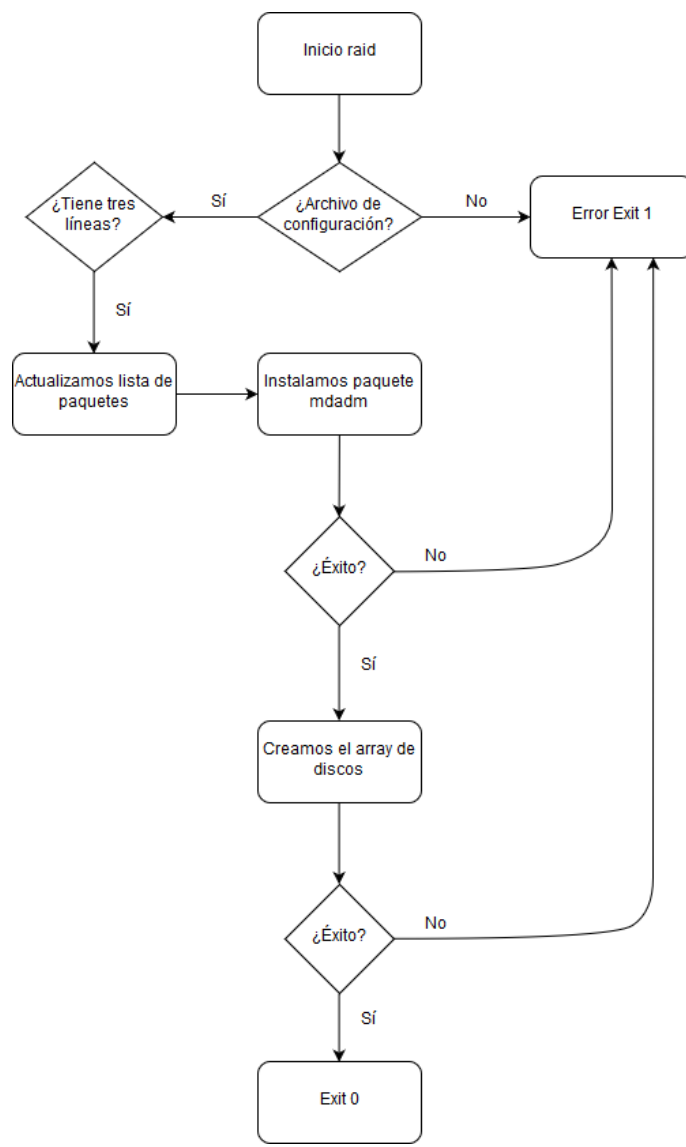
Como comprobaciones generales a lo largo del script se realiza:

- Que se haya especificado un archivo de configuración
- Que el número de líneas del archivo de configuración sea de tres líneas
- Que la lista de paquetes esté actualizada

Una vez realizadas las comprobaciones se procederá a instalar el paquete mdadm y posteriormente se creará el array de discos. En caso que los discos indicados no existan o se encuentren ocupados por otro proceso, la creación del array fallará y se abortará la ejecución.

Referente a las dificultades, hemos encontrado una al instalar el paquete mdadm, en el que nos salía una pantalla que necesitaba interacción para establecer el nombre del raid. La solución empleada ha sido poner DEBIAN_FRONTEND=noninteractive en el comando apt.

A continuación, se muestra el diagrama que resume el funcionamiento de dicho script:



2.3 LVM

El programa en el que está desarrollado este servicio es lvm.sh
Como comprobaciones generales a lo largo del script se realiza:

- Nº líneas del fichero de configuración que leemos >3
- Comprobar que los paquetes del lvm2 están instalados
- Suma del tamaño de los dispositivos sea > 0
- Existan los dispositivos que se van a utilizar en el sistema
- Tamaño del volumen a crear < Tamaño disponible.
- En todo momento se comprueba que cada operación que realiza el sistema termina correctamente.
-

Si no se cumple lo anterior el programa termina con exit 1.

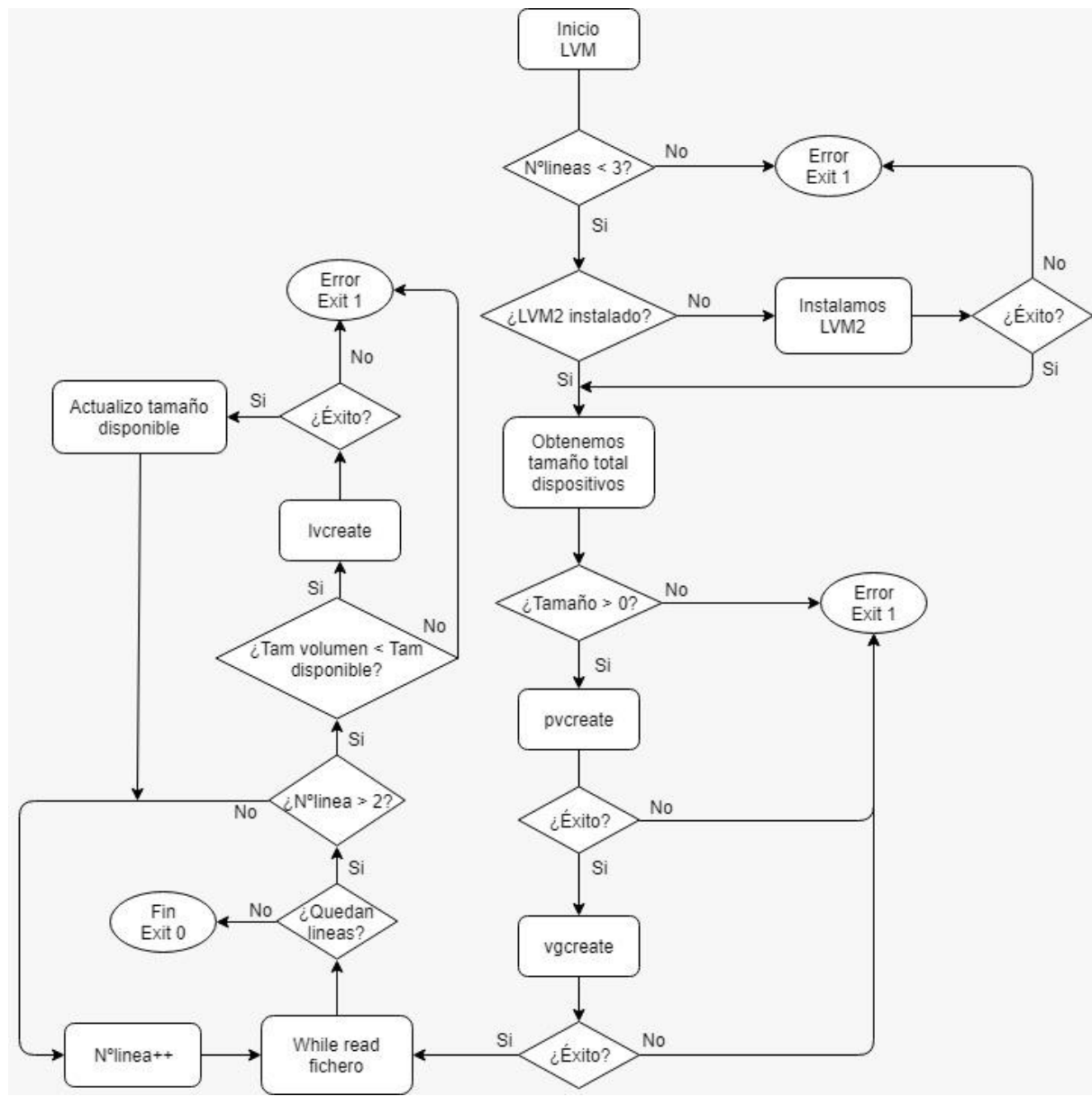
Como dificultades encontradas en la realización de este servicio ha sido la comprobación de si estaban instalados los paquetes de lvm2, que finalmente se hizo con esta sentencia, viendo si terminaba correctamente:

```
dpkg -l | grep "^ii" | awk '{ print $2 }' | grep "^lvm2$"
```

También para poder coger el tamaño de cada dispositivo en la misma unidad(GB) y sin el GB del final, que se hizo de esta forma:

```
tamGB=$(lsblk ${listadisp[i]} -b -o SIZE | sed -n 2p | awk '{ byte =$1 /1024/1024/1024 ; print  
byte "GB" }')  
tamReal=${tamGB::-2}
```

A continuación se muestra un diagrama que resume el funcionamiento de dicho script:



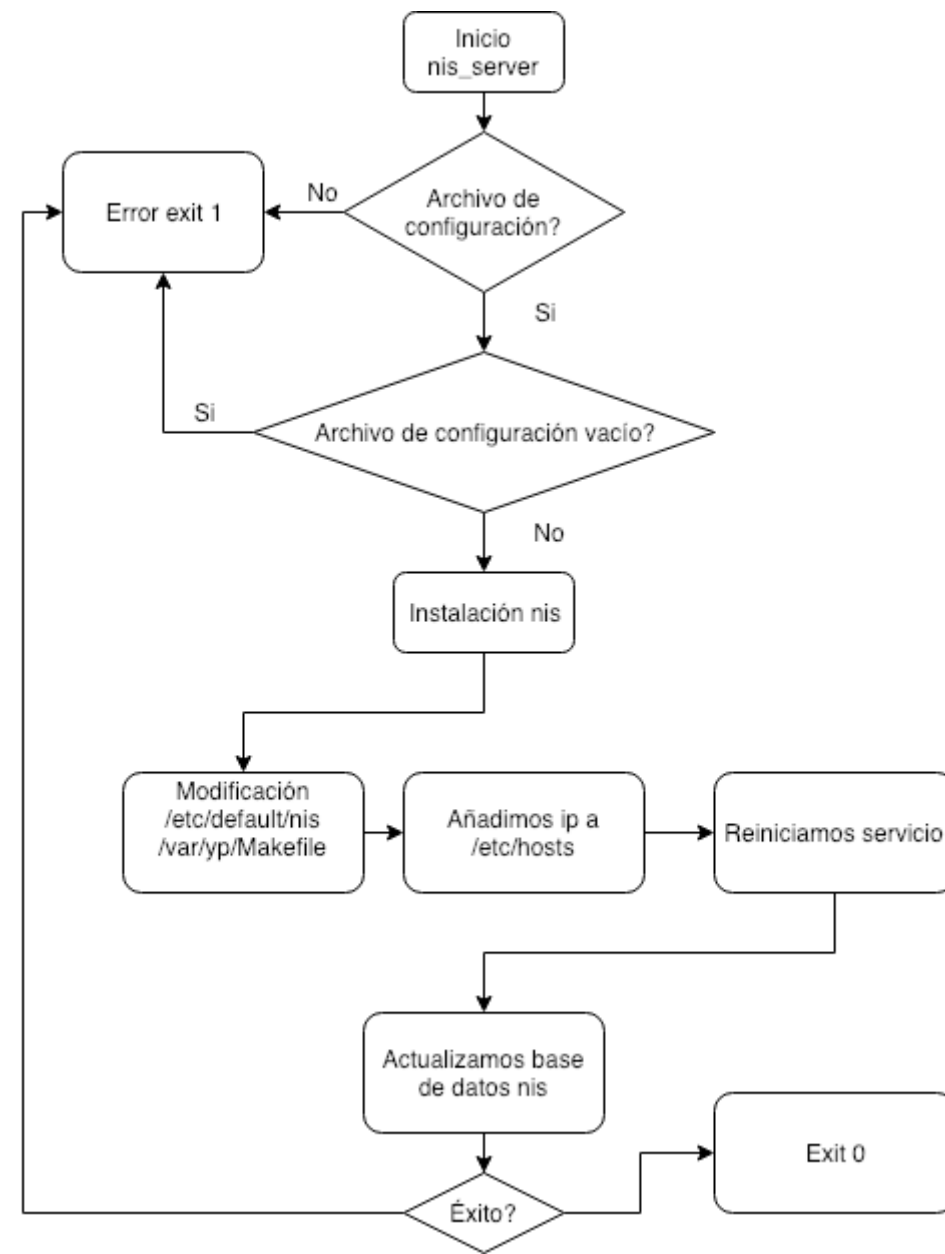
2.4 NIS Server

El programa en el que está desarrollado este servicio es nis_server.sh.

Como comprobaciones generales a lo largo del script se realiza:

- Que se haya especificado un archivo de configuración.
- Dominio especificado para nis.

A continuación se muestra un diagrama que resume el funcionamiento de dicho script:



Con este script tuvimos varias complicaciones, una de ellas fue el que se tenía que ejecutar de manera que no tuviera interacción con el usuario, puesto que al ejecutarlo así configuraba un nombre de dominio diferente al que se especificaba en el archivo de configuración.

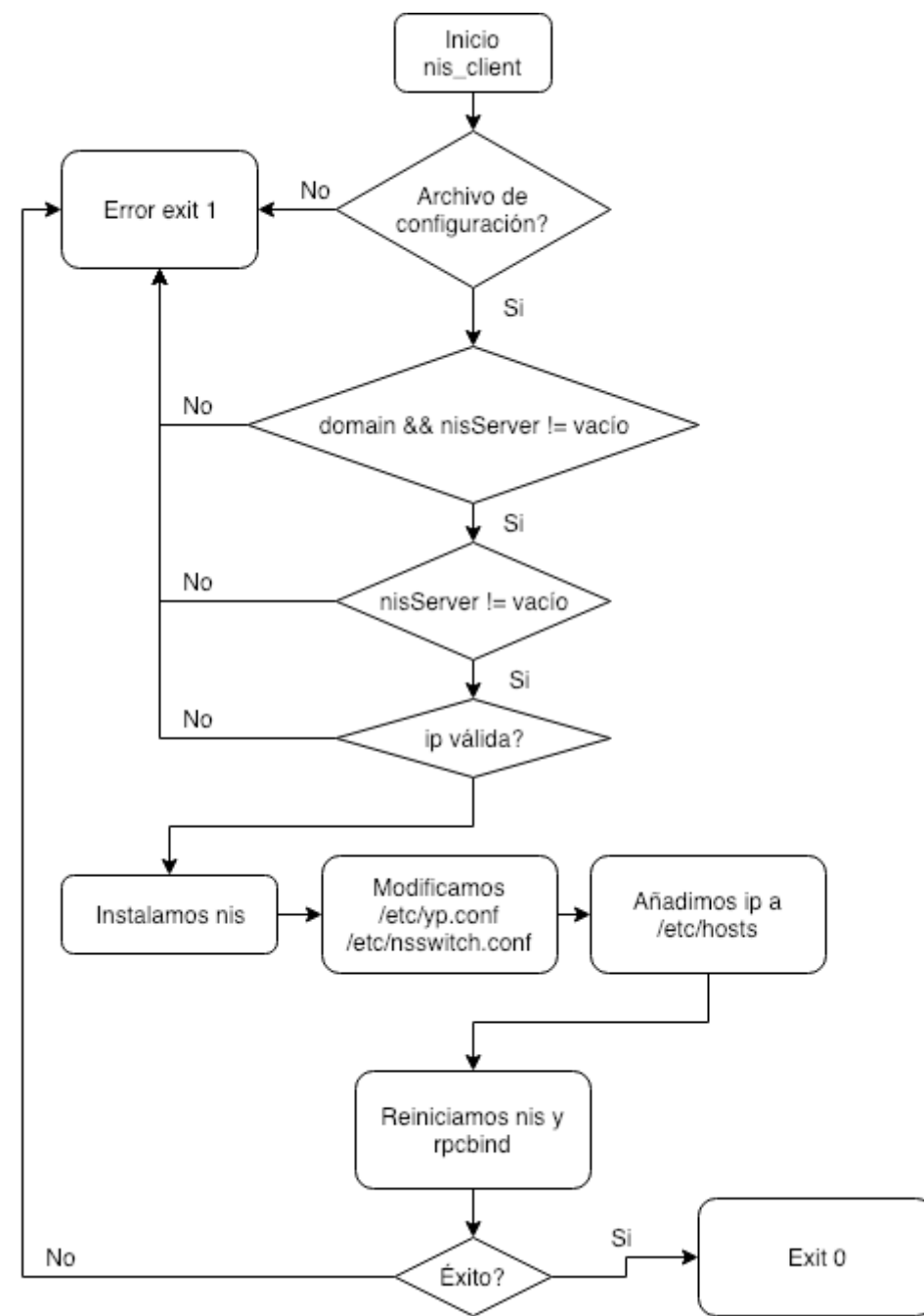
2.5 NIS Client

El programa en el que está desarrollado este servicio es nis_client.sh.

Como comprobaciones generales a lo largo del script se realiza:

- Que se haya especificado un archivo de configuración.
- Dominio nis y servidor nis al cual se requiere conectar.
- Servidor nis válido.

A continuación se muestra un diagrama que resume el funcionamiento de dicho script:



Las dificultades encontradas en este script son las mismas que se tuvieron a la hora de realizar el script de `nis_server`, puesto que al no poder configurar de manera adecuada el servidor, los clientes no se podían conectar a este, por lo que nos impedía avanzar con este script si todavía no quedaba el anterior.

2.6 NFS Server

El programa en el que está desarrollado este servicio es `nfs_server.sh`

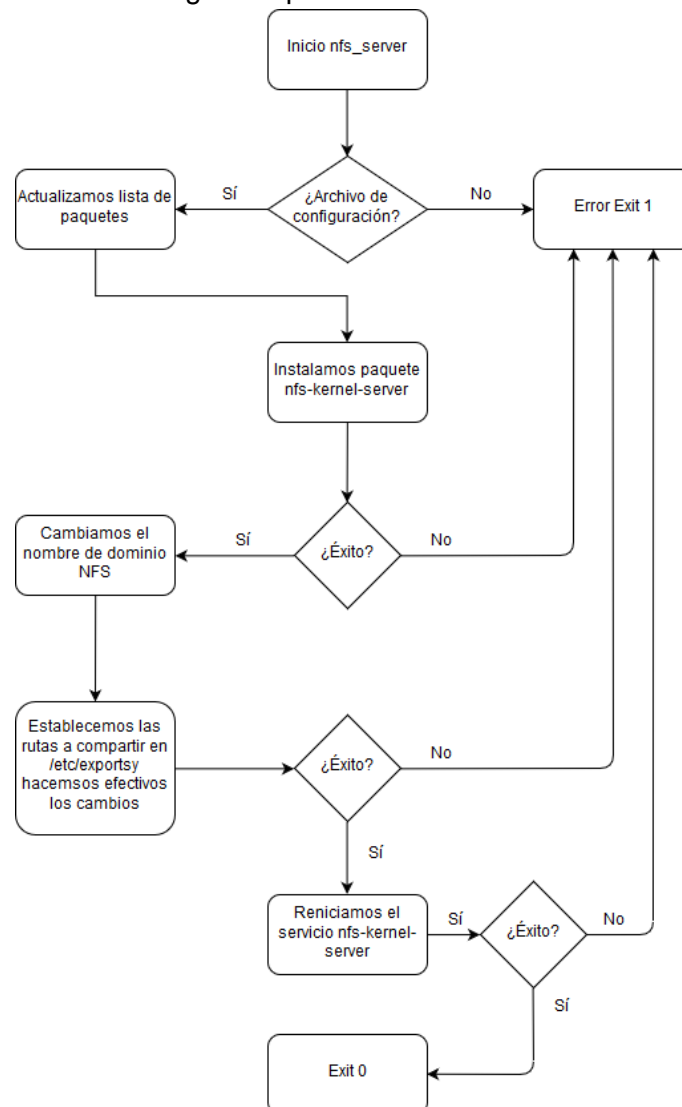
Como comprobaciones generales a lo largo del script se realiza:

- Que se haya especificado un archivo de configuración
- Que la lista de paquetes esté actualizada
- Que las rutas a compartir se añaden correctamente en `/etc/exports`
- Que se reinicia el servicio `nfs-kernel-server` correctamente

Una vez realizadas las comprobaciones se procederá a instalar el paquete `nfs-kernel-server` y posteriormente se realizará la configuración del servidor, que consistirá en cambiar en nombre de dominio del NFS y establecer las rutas de los directorios que se compartirán.

En cuanto a las dificultades, la única que se ha encontrado ha sido la inclusión de una nueva línea en un archivo y que realizaba de manera errónea al faltar el flag `-i` en el comando `sed`.

A continuación, se muestra el diagrama que resume el funcionamiento de dicho script:



2.7 NFS Client

El programa en el que está desarrollado este servicio es `nfs_client.sh`

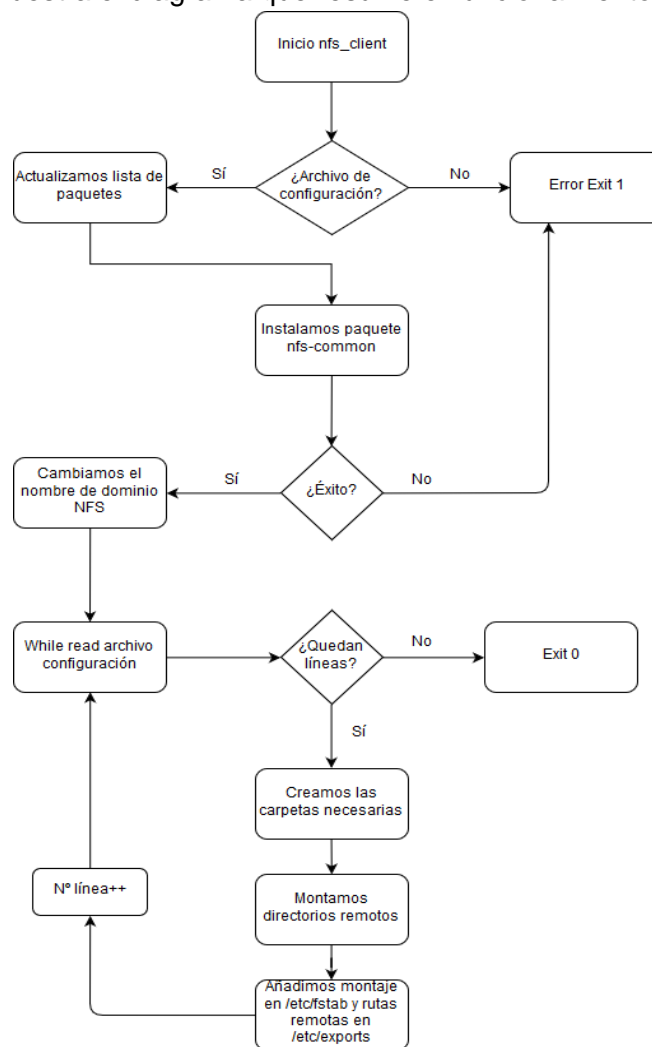
Como comprobaciones generales a lo largo del script se realiza:

- Que se haya especificado un archivo de configuración
- Que la lista de paquetes esté actualizada
- Que se instala el paquete `nfs-common`
- Que se montan correctamente los directorios remotos

Una vez realizadas las comprobaciones iniciales se procederá a instalar el paquete `nfs-common` y posteriormente se realizará la configuración del cliente, que consistirá en cambiar en nombre de dominio del NFS, en crear las carpetas, en el caso de que no existan, donde se montarán los directorios remotos, en montar dichos directorios y añadirlos al fichero `/etc/fstab` para que se monten automáticamente cada vez que arranque el sistema y en establecer las rutas de los directorios que se compartirán en `/etc/exports`.

Una vez comprendido el proceso de configuración del cliente no se han encontrado dificultades.

A continuación, se muestra el diagrama que resume el funcionamiento de dicho script:



2.8 Backup Server

La tecnología que se ha decidido utilizar en la práctica es rsync, ya que es el método que se ve en clase y más sencillo de incorporar.

El programa en el que está desarrollado este servicio es backup_server.sh

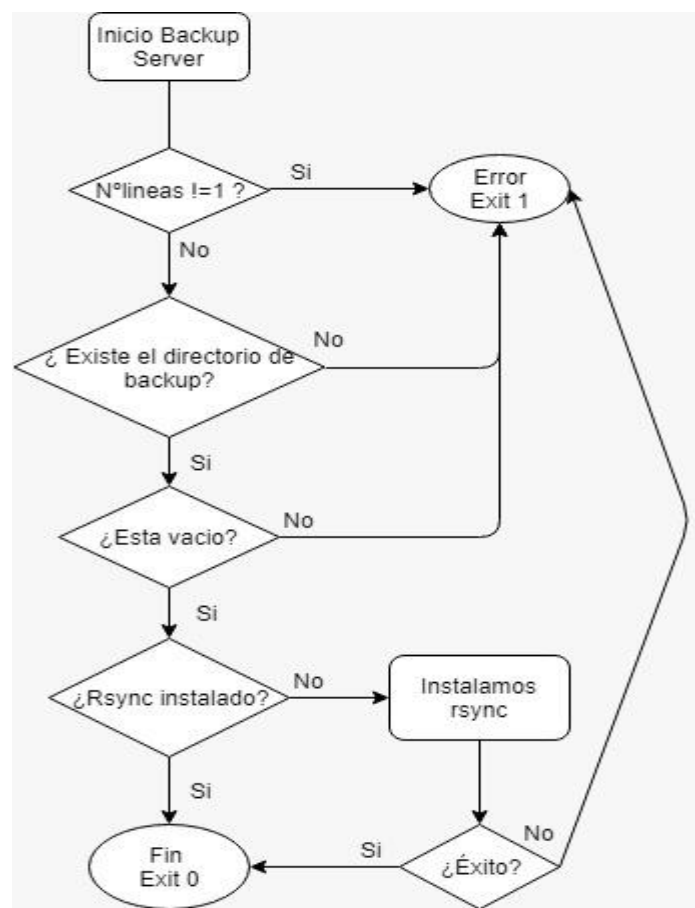
Como comprobaciones generales a lo largo del script se realiza:

- Nº líneas del fichero de configuración que leemos = 1.
- Comprobar que el directorio de backup existe y está vacío.
- Comprobar que rsync está instalado.
- En todo momento se comprueba que cada operación que realiza el sistema termina correctamente.

Si no se cumple lo anterior el programa termina con exit 1.

En cuanto a dificultades encontradas en este script no ha surgido ninguna.

A continuación se muestra un diagrama que resume el funcionamiento de dicho script:



2.9 Backup Client

El programa en el que está desarrollado este servicio es backup_client.sh.

Como comprobaciones generales a lo largo del script se realiza:

- Nº líneas del fichero de configuración que leemos = 4
- Comprobar que el directorio local del que se quiere hacer backup existe.
- Comprobar que el directorio de destino del servidor existe y tenemos permisos de escritura.
- Comprobar que rsync está instalado.
- En todo momento se comprueba que cada operación que realiza el sistema termina correctamente.

Si no se cumple lo anterior el programa termina con exit 1.

En cuanto a dificultades encontradas en este script no ha surgido ninguna.

Comentar que en el mandato rsync se ha indicado las opciones para que copie todo el contenido del directorio (enlaces simbólicos, permisos ...), además se comprima cuando se realice el backup para que ocupe poco, y que el próximo backup solo copie los ficheros que se han modificado.

A continuación se muestra un diagrama que resume el funcionamiento de dicho script:

