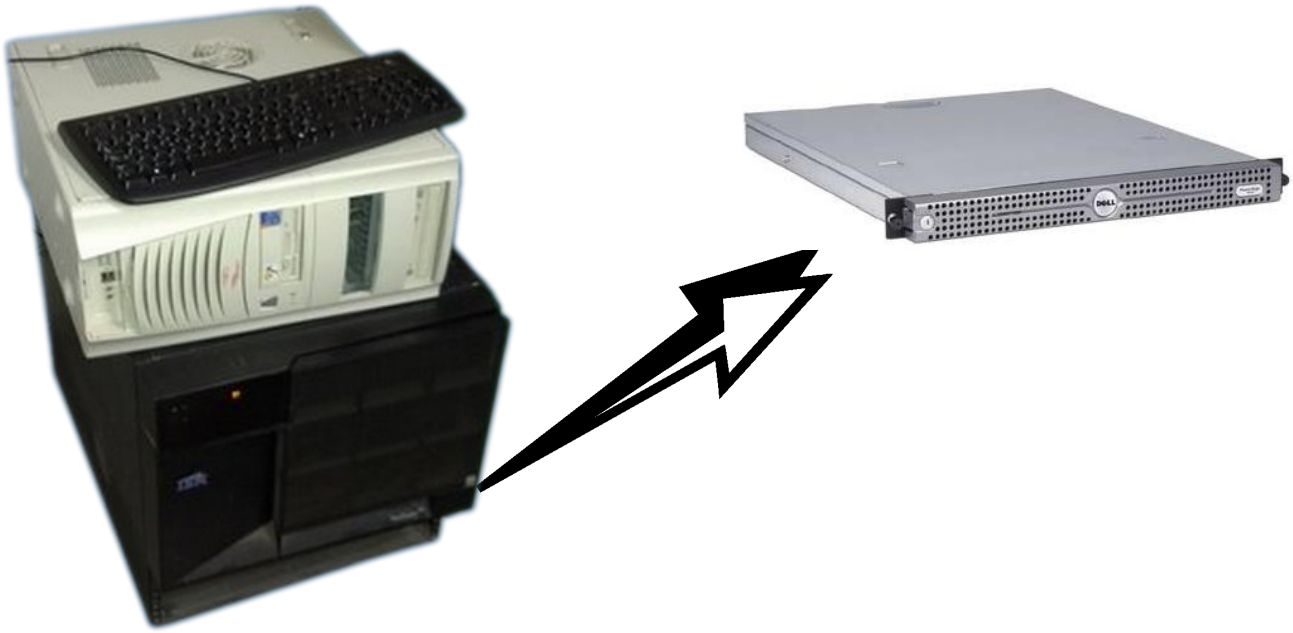
MIGRACIÓN DE SERVIDORES



Proyecto Integrado de ASI
Realizado por: Jesús Lucas Flores Curso 2º ASI
Junio 2010 IES Gonzalo Nazareno de Dos Hermanas

INDICE

Índice de contenido

1.- Introducción

1.1 Presentación

La realización de este proyecto implica conocer las dificultades y problemas que surgen al realizar de forma práctica la migración de dos servidores que se encuentran en producción a un sistema nuevo sin perder nada de información o perdiendo el mínimo de información posible, también se verá reflejado en este las mejoras posibles una vez realizada la migración.

Este proyecto es realizado por el alumnos Jesús Lucas con la ayudar y colaboración de su tutor de dicho proyecto Alberto Molina.

1.2 Objetivos

Los objetivos de este proyecto inicialmente son lo siguientes, debiendo destacar que pueden sufrir modificaciones y ampliaciones durante el transcurso del desarrollo del mismo.

• Comprender la necesidad de migrar servidores en una situación real.

• Realizar la migración de los servidores “Lavadora” y “Aduana” del IES Gonzalo Nazareno a un sistema de máquinas virtuales KVM y resolver todos los problemas que surjan en el transcurso de esta.

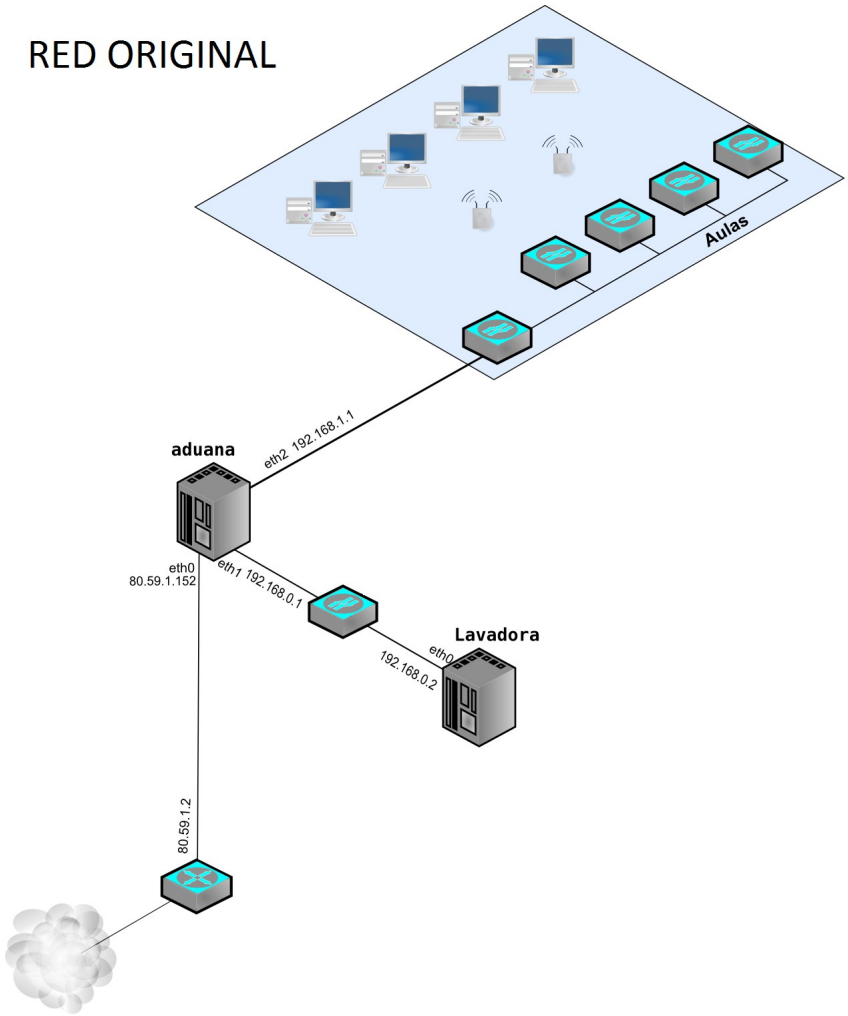
• Analizar y proponer mejoras posibles una vez realizada la migración.

Como objetivo adicionales se aprenderá a:

•Aprender Latex Básico y colaborar en el desarrollo de la documentación del nuevo servidor.

2.- Situación actual.

2.1 Análisis de la situación inicial y sus problemas.



Como podemos ver en la imagen la situación antes de la migración tiene diversas problemáticas.

La red se encuentra dividida de la siguiente forma:

Red de Aulas: 192.168.1.0/24
DMZ: 192.168.0.0/24

Como puerta de enlace a Internet se dispone de un router configurado en mono-puesto que pasa su ip pública a la interfaz eth1 de aduana.

Aduana es el firewall de toda la red y la puerta de enlace de todos los equipos del aula y de la DMZ.

Lavadora es un servidor de muy baja eficiencia debido a sus características. Es un servidor IBM

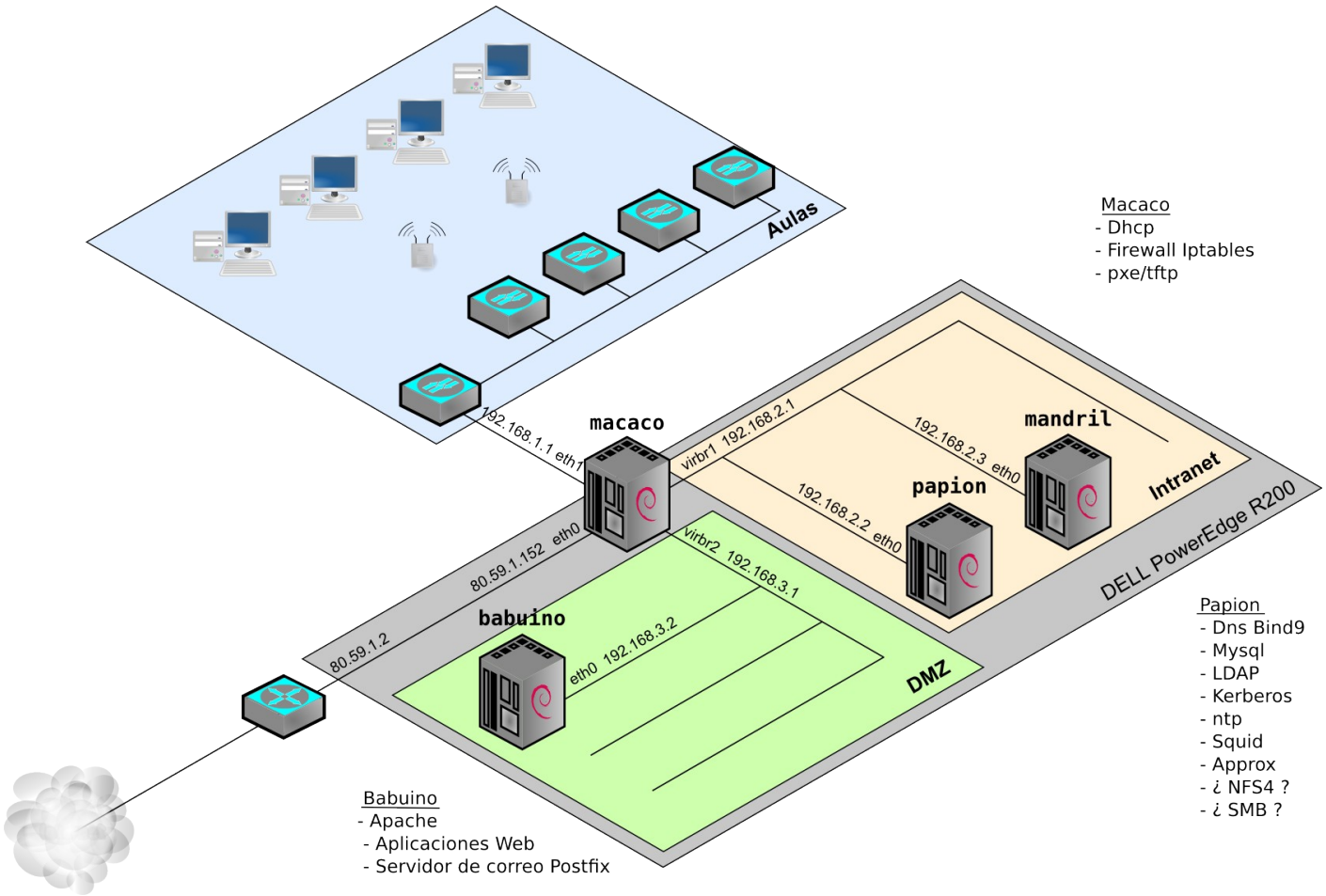
Netfinity 5500 con las siguientes características principales:

- 2 CPU Pentium III 600 MHz
- 512 MiB RAM
- Controladora RAID Hardware

El principal problema de la red es la capacidad de procesamiento y la memoria de estos equipos, lo que provoca que se ralentice debido a las aplicaciones web, bases de datos y otros servicios, llegando incluso a colgarse en algunas ocasiones.

El otro gran problema de la red es la capacidad de subida del ADSL contratado, por lo que es necesario instalar un servicio de QOS para dar prioridad a cierto tipo de paquetes. De solventar este problema se encargará Carlos Álvarez que será el encargado de la implantación del mismo.

2.2 Ventajas con la migración.



Con la migración que se realizará se pretenderá solucionar los problemas que anteriormente se nombraron.

En la imagen podemos ver la situación final de la migración:

Solo se usara un solo equipo físico para los servidores, un servidor Dell PowerEdge R200 con las

siguientes características principales:

◦ Procesador Dual Core Intel Xeon E3120, 3.16GHz, 6MB Cache, 1333MHz FSB.
◦ 8 GiB de RAM, DDR2 a 800 MHz (4x2GiB).
◦ 3 NIC Broadcom Corporation NetXtreme BCM5721 (Gigabit).
◦ Unidad DVD+/-RW SATA.
◦ 2 discos duros SATA de 500 GiB a 7200 rpm
◦ Controladora SATA SAS6iR configurada en RAID1 (hardware)

Este servidor llamado "**macaco**" dispondrá de un Debian Lenny con KVM configurado, para la creación del sistema de máquinas virtuales Debian sobre el cual estará basado la nueva red.

La red quedará dividida en las siguientes subredes:

Aulas: 192.168.1.0/24
Intranet: 192.168.2.0/24
DMZ: 192.168.3.0/24

En la Intranet se creará una máquina virtual "**papion**":
- Ip: 192.168.2.2

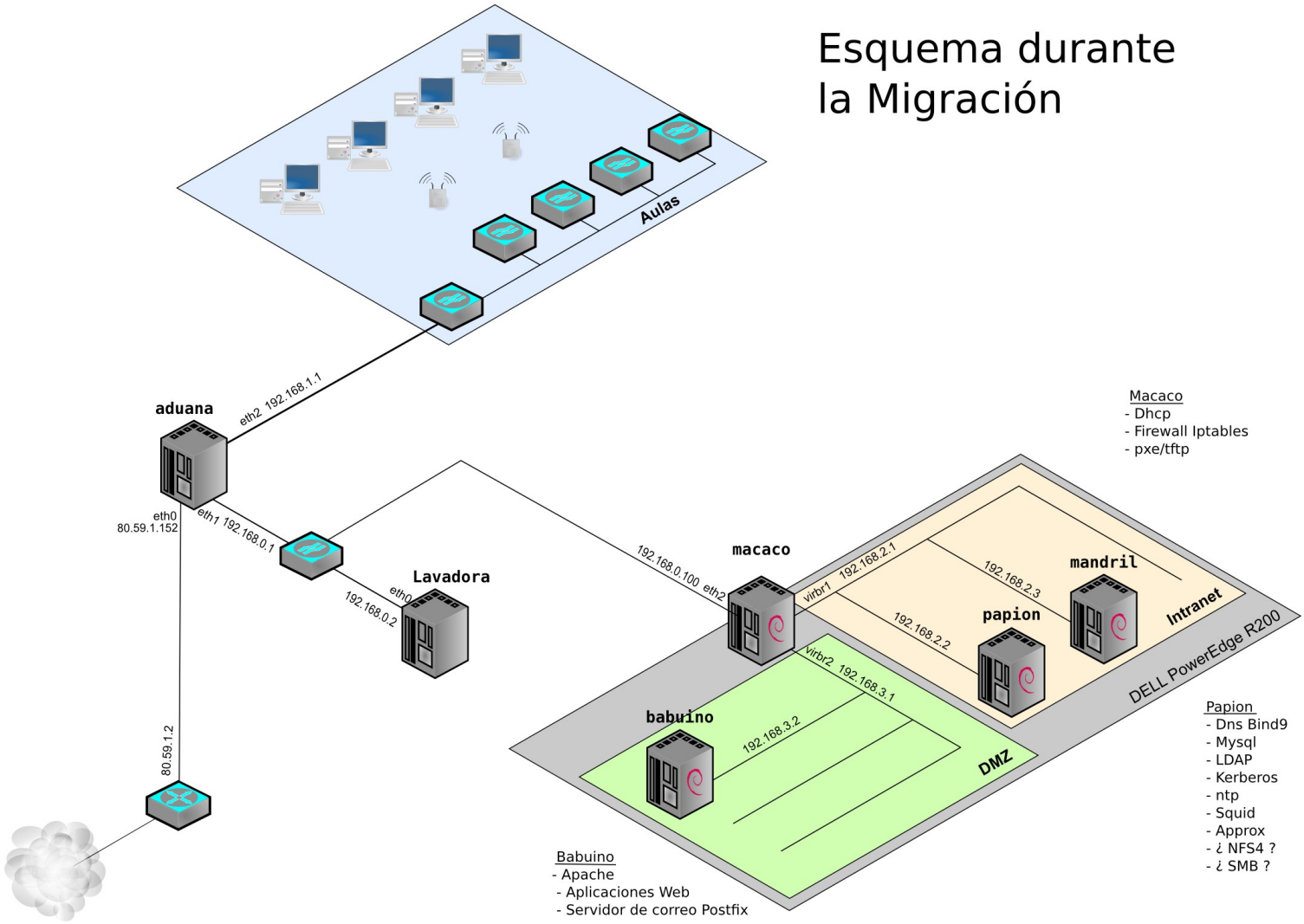
Papion contendrá los servicios que solo deben ser accesibles desde 192.168.0.0/16 que serán los que podemos ver en la imagen.

Por otro lado dispondremos de un equipo en la DMZ, "**babuino**":
- Ip: 192.168.3.2

Babuino tendrá aquellos servicios que son accesibles desde internet.

El servidor "macaco" tendrá el servidor DHCP y el firewall Iptables.

3.- Marco de desarrollo durante la migración.

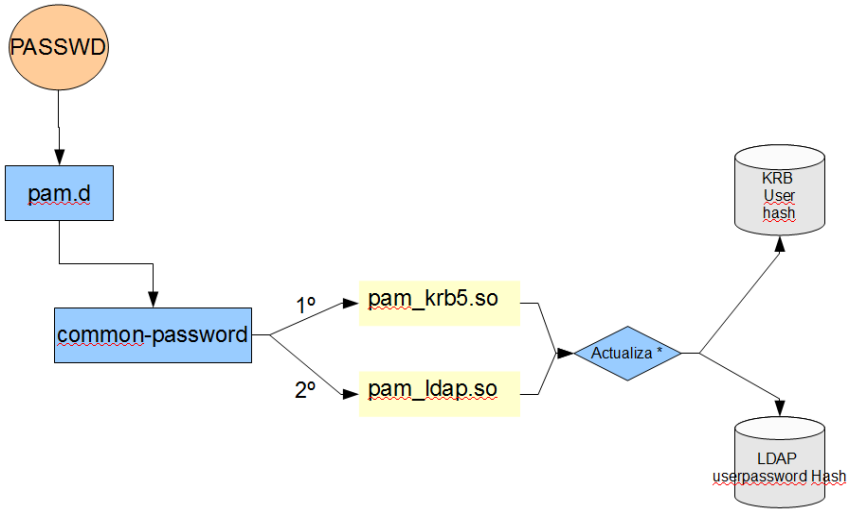


Durante el proceso de migración de los servidores la estructura de la red será la que podemos ver en la imagen. Lavadora y aduna seguirán en pleno funcionamiento mientras se realizan las migraciones correspondientes.

La migración se podrá llevar acabo desde el exterior accediendo al servidor Macaco por medio de ssh a la dirección informatica.gonzalonazareno.org por el puerto 2222 que nos pondrá en contacto para acceder por medio de ssh a las máquinas virtuales.

4.- Investigando: Sincronización de Hashes Kerberos-LDAP

La situación ideal a la que desearíamos llegar sería la siguiente:



* Actualiza:
Primeramente realizara el update sobre KRB y posteriormente en la entrada LDAP correspondiente.

Una vez encontrada la forma de realizarlo se podría realizar la implantación de alguna aplicación web desarrollada específicamente para el cambio de contraseñas y/o creación de usuarios, este objetivo se escapa del ámbito de este proyecto por lo tanto lo propongo para la realización de proyectos futuros.

Se creará un entorno de pruebas con dos máquinas Debian.

Una máquina actuará como servidor Kerberos con LDAP y la otra será el cliente de esta.

Usar las librerías: pam_krb5.so y pam_ldap.so.

Editar el fichero common-password encargado de la actualización de las contraseñas de manera que actualice primero Kerberos y luego LDAP o viceversa.

Entorno de Pruebas:

Dos máquinas Debian Lenny:

```
Jack:
192.168.1.103
dominio aceclubs.org
```

```
King :
192.168.1.102
Servidor KRB, LDAP, DNS , NTP
LDAP : dc=aceclubs,dc=org
Kerberos: hay principal pruebau
dominio aceclubs.org
```

Disponemos el entorno configurado para que jack sea cliente kerberos de king y tiene su pam (Pluggable authentication module) configurado para permitir el login de los usuarios de kerberos que dispone King.

Ahora procederemos a experimentar y realizar pruebas hasta intentar conseguir o acercarnos lo más posible al objetivo antes definido:

Logueados como pruebau en jack probamos cambiar su password, teniendo los ficheros pam configurados para el login mediante kerberos.

Con esta configuración de los ficheros pam, se pide el password Kerberos y se cambia. Lo hemos cambiado a pruebau. Pero no se cambia en el servidor LDAP.

- Añadido entrada user_password a cn=pruebau
- Hemos hecho un dpkg-reconfigure de libpam-ldap y como cifrado elegimos EXOP.

/etc/pam.d/common-password

```
password   sufficient pam_ldap.so md5
password   required   pam_unix.so nullok obscure md5
```

Con este common-password al hacer:

```
pruebau@jack:/$ passwd
```

Nos pide contraseña de LDAP .

/etc/pam.d/common-password

```
password   sufficient pam_ldap.so md5
password   required   pam_unix.so nullok obscure md5
```

Con este "*common-password"* al hacer:

```
pruebau@jack:/$ passwd
```

```
 Current Kerberos password:
```

```
Enter  login(LDAP) password:
Enter new Kerberos password:
Retype new Kerberos password:
New password:
Re-enter new password:
     LDAP password information changed for pruebau
      passwd: contraseña actualizada correctamente.
```

Muy bien con esto hacemos que el comando passwd nos actualice Kerberos y LDAP aunque debemos introducir la contraseña actual dos veces y actualiza por separado.

Podríamos desarrollar un Script en php que use este passswd y cambiar la contraseña desde Drupal donde solo se introdujera la antigua contraseña y la nueva.

Para el desarrollo de este script de podría usar la siguiente información:

http://bash.cyberciti.biz/security/change-password-shell-script/

5.- Migración de Servicios y Aplicaciones.

5.1 Migrando NTP y OpenLDAP en Papion

a) Instalando el Servidor de Hora NTP.

Instalamos el paquete ntp:

```
papion:~# aptitude install ntp
```

que se conecta a servidores de hora públicos de Internet para sincronizar su reloj y además ofrece el servicio ntp (123/udp) al resto de equipos de la red. Tras iniciar el demonio, se pueden ver en el fichero /var/log/syslog registros como:

```
ntpd[32172]: synchronized to 213.97.131.125, stratum 2 3
ntpd[32172]: kernel time sync status change 0001
ntpd[32172]: synchronized to 147.83.123.136, stratum 2
```

Se ofrece el servicio ntp por todas las interfaces disponibles:

```
papion:~# netstat -putan |grep ntp
```

```
udp                 0             0  192.168.2.2:123                    0.0.0.0:*
1986/ntpd
udp                 0             0  127.0.0.1:123                      0.0.0.0:*
1986/ntpd
udp                 0             0  0.0.0.0:123                        0.0.0.0:*
1986/ntpd
udp6                       0                  0   fe80::216:36ff:fe43:123   :::*
1986/ntpd
udp6                0             0  ::1:123                            :::*
1986/ntpd
udp6           0           0 :::123                                      :::*
1986/ntpd
```

b) Instalacion de OpenLDAP.

- **Configuración del Hostname.**

En principio:

```
papion:~# hostname
papion
papion:~# hostname -f
hostname: Unknown host
```

Donde Unknown host debería ser el FQDN de papion: papion.gonzalonazareno.org

Editando el fichero /etc/hosts , añadimos la linea segunda:

```
127.0.0.1 localhost
127.0.1.1 papion.gonzalonazareno.org papion
```

Ahora:

```
papion:~# hostname
papion
papion:~# hostname -f
papion.gonzalonazareno.org
```

- **Instalando Slapd.**

En primer lugar debemos instalar el paquete slapd y todas sus dependencias:

```
papion:~# aptitude install slapd
```

A continuación (dependiendo de la configuración del paquete debconf) nos pedirá lo siguiente:

Contraseña del Administrador del directorio

Se introduce la misma que tiene ahora mimos el root de esta máquina y creará un directorio con dos entradas, en nuestro caso:

```
papion:~# slapcat
```

```
dn: dc=gonzalonazareno,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
o: gonzalonazareno.org
dc: gonzalonazareno
structuralObjectClass: organization
entryUUID: e0fbb652-e5ab-102e-98cc-51f05adfa456
creatorsName:
createTimestamp: 20100426181838Z
entryCSN: 20100426181838.044978Z#000000#000#000000
modifiersName:
modifyTimestamp: 20100426181838Z

dn: cn=admin,dc=gonzalonazareno,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e2NyeXB0fVd0aDB4Qy5SQlZ1OVk=
structuralObjectClass: organizationalRole
entryUUID: e0fbf8e2-e5ab-102e-98cd-51f05adfa456
creatorsName:
createTimestamp: 20100426181838Z
entryCSN: 20100426181838.046805Z#000000#000#000000
modifiersName:
modifyTimestamp: 20100426181838Z
```

Vamos a reconfigurar el paquete ya que la base no se corresponde con la que necesitamos:

```
dpkg-reconfigure slapd
```

Nombre de dominio DNS: gonzalonazareno.org
Nombre de la Organización: gonzalonazareno.org
Contraseña del administrador (la misma que la anterior)
Motor de base de datos a utilizar: BDB
¿Permitir el protocolo LDAPv2?: No

Ahora tenemos las siguiente entradas:

```
papion:~# slapcat
```

```
dn: dc=gonzalonazareno,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
o: gonzalonazareno.org
dc: gonzalonazareno
structuralObjectClass: organization
entryUUID: 518f3eb4-e5b3-102e-84ac-2b9d7b066ddc
creatorsName:
createTimestamp: 20100426191153Z
entryCSN: 20100426191153.393770Z#000000#000#000000
modifiersName:
modifyTimestamp: 20100426191153Z
dn: cn=admin,dc=gonzalonazareno,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e2NyeXB0fWc2WXNvTzBGWnN4amc=
structuralObjectClass: organizationalRole
entryUUID: 518f890a-e5b3-102e-84ad-2b9d7b066ddc
creatorsName:
createTimestamp: 20100426191153Z
entryCSN: 20100426191153.395801Z#000000#000#000000
modifiersName:
modifyTimestamp: 20100426191153Z
```

- **Configuración cliente ldap de papion.**

Editamos el archivo /etc/ldap/ldap.conf y lo configuramos de la siguiente manera:

```
papion:~# nano /etc/ldap/ldap.conf
```

```
# This file should be world readable but not world writable.
BASE  dc=gonzalonazareno,dc=org
URI   ldap://papion.gonzalonazareno.org

#SIZELIMIT  12
```

```
#TIMELIMIT  15
#DEREF          never
```

- **Instalando las herramientas ldap-utils.**

Instalamos las herramientas ldap-utils tan sencillo como:

```
papion:~# aptitude install ldap-utils
```

- **Migrando el LDAP de LAVADORA.**

Dos opciones:

A) Exportamos el árbol de lavadora a un fichero ldif y luego lo añadimos a nuestro ldap de papion.

- Necesito permisos para usar slapcat y pasar la base de datos a un fichero ldif.

B) Copiamos la carpeta de lavadora donde se encuentra la base de datos físicamente: /var/lib/ldap y los pegamos en la misma ruta pero en papion.

- Supongo que para realizar la copia el demonio slapd de lavadora debería de pararse unos instantes.
- Necesito permiso recursivo de lectura sobre la carpeta /var/lib/ldap de lavadora.

Decidimos optar por la opción más sencilla, eficiente y práctica: opción A

Disponemos ya de nuestro fichero Ldif "lavadora2.ldif" con el volcado del LDAP de lavadora en macaco en el home de root: /root

Pasaremos este archivo al home de root de papion con scp.

```
macaco# scp /root/lavadora2.ldif root@192.168.2.2:/root
root@192.168.2.2's password:
lavadora2.ldif                    100%   87KB  87.4KB/s   00:00
```

Una vez tenemos el fichero en papion debemos parar antes el demonio slapd antes de cargar el fichero ldif y luego cargarlo:

```
papion# /etc/init.d/slapd stop
papion# slapadd -l lavadora2.ldif
papion# /etc/init.d/slapd start
```

- **Indices LDAP**

Una vez que el directorio está funcionando y recibiendo peticiones, aparecen habitualmente registros en el fichero /var/log/syslog del tipo:

```
Sep 11 20:00:21 lavadora slapd[11449]: <= bdb_equality_candidates: (uid) not
indexed
```

que nos recomiendan indexar determinados atributos, en este caso uid, para que editar el fichero de configuración /etc/ldap/slapd.conf y añadir la linea:

```
index       uid     eq
```

Y actualizar los indices con el directorio parado:

```
papion:~#    /etc/init.d/slapd stop
papion:~#    slapindex
papion:~#    chown openldap slapd.conf
papion:~#    /etc/init.d/slapd start
```

- **ACL de profesores**

Para el correcto funcionamiento de la aplicación Gestiona los usuarios que pertenezcan al grupo profesores deben tener permiso de escritura sobre todos los atributos del directorio, para así poder crear o modificar usuarios.

En primer lugar hay que añadir los uid de todos los miembros del grupo profesores.

Esto ya se encuentra realizado ya que importamos el árbol de lavadora donde ya estaba hecho.

Por lo que pasamos a realizar la siguiente modificación en las ACL para que los miembros del grupo profesores sean administradores del directorio:

```
access to attrs=userPassword,shadowLastChange
      by dn="cn=admin,dc=gonzalonazareno,dc=org" write
      by set="([uid=] + ([cn=profesores,ou=Group,dc=gonzalonazareno,\
      dc=org])/memberUid + [,ou=People,dc=gonzalonazareno,dc=org])\
      /entryDN & user" write
      by anonymous auth
      by dn="cn=usuario-nss,dc=gonzalonazareno,dc=org" read
      by self write
      by * none
access to *
      by set="([uid=] + ([cn=profesores,ou=Group,dc=gonzalonazareno,\
      dc=org])/memberUid + [,ou=People,dc=gonzalonazareno,dc=org])\
      /entryDN & user" write
      by group="cn=profesores,ou=Group,dc=gonzalonazareno,dc=org" write
      by dn="cn=usuario-nss,dc=gonzalonazareno,dc=org" read
      by self write
      by * read
```

Para que el usuario usuario-nss tenga permiso de lectura sobre todo el árbol se añadió la linea:

```
by dn="cn=usuario-nss,dc=gonzalonazareno,dc=org" read
```

Reiniciamos el servicio:

```
papion:~#    /etc/init.d/slapd restart
```

- Inst**alando librerías LDAP necesarias para Kerberos+LDAP**

A continuación se instalan los paquetes libnss-ldap poniendo "cn=usuario-nss,dc=gonzalonazareno, dc=org" cuando se solicite el usuario de conexión al directorio.

En esta configuración concreta no es necesario que el usuario anterior tenga permiso de escritura sobre el directorio LDAP porque las contraseñas no se van a modificar con pam ldap.

```
ldapsearch -x -D "cn=usuario-nss,dc=gonzalonazareno,dc=org" -W
```

Probamos la contraseña de usuario-nss.

```
Papion:~# aptitude install libnss-ldap libpam-ldap nscd
```

Pondremos la contraseña cuando nos diga que si se necesita un usuario para leer la base de datos daremos a Si.

5.2 Migrando el Servidor DNS:

Antes de pasar a la instalación de Kerberos realizaremos la importación del servidor DNS de aduana papion. Instalando Bind

- Instalando Bind.

```
Papion:~# aptitude install bind9
Papion:~# aptitude install dnsutils
```

- Migrando Bind de Aduana a Papion

Entramos a aduana copiamos los ficheros de configuración.
Importaremos los siguientes archivos de Dns de aduana:

Configuración: /etc/bind

```
-rw-r--r-- 1 root bind 1023 oct 26  2009 named.conf
-rw-r--r-- 1 root bind  401 dic  2 10:21 named.conf.local
-rw-r--r-- 1 root bind  598 nov 23 10:48 named.conf.options
```

Zonas: /var/cache/bind

```
-rw-r--r-- 1 bind bind     720 abr 28 17:54 db.192.168
-rw-r--r-- 1 bind bind     914 abr 28 17:51 db.gonzalonazareno
```

Para ello paramos el servicio bind de papion:

```
Papion:~# /etc/init.d/bind9 stop
```

y pasamos los ficheros por medio de ssh con scp;

```
Papion:~# scp jesuslucas@192.168.0.1:/etc/bind/named.conf /etc/bind
```

```
Papion:~# scp jesuslucas@192.168.0.1:/etc/bind/named.conf.local /etc/bind
Papion:~# scp jesuslucas@192.168.0.1:/etc/bind/named.conf.options /etc/bind
Papion:~# scp jesuslucas@192.168.0.1:/var/cache/bind/db.192.168 /var/cache/bind
Papion:~#     scp     jesuslucas@192.168.0.1:/var/cache/bind/db.gonzalonazareno
/var/cache/bind
```

Editamos el fichero /var/cache/bind/db.192.168 con los datos actuales dejándolo así:

```
papion:/var/cache/bind# nano db.192.168
```

```
$ORIGIN
$TTL 86400      ; 1 day
168.192.in-addr.arpa    IN SOA  papion.168.192.in-addr.arpa. Postmas$
23413      ; serial
21600      ; refresh (6 hours)
3600       ; retry (1 hour)
604800     ; expire (1 week)
21600      ; minimum (6 hours)
)
NS      papion.gonzalonazareno.org.
$ORIGIN 0.168.192.in-addr.arpa.
10                      PTR     macaco.gonzalonazareno.org.
2                       PTR     lavadora.gonzalonazareno.org.
$ORIGIN 1.168.192.in-addr.arpa.
1                       PTR     aduana.gonzalonazareno.org.
$ORIGIN 2.168.192.in-addr.arpa.
2                       PTR     papion.gonzalonazareno.org.
3                       PTR     mandril.gonzalonazareno.org.
$ORIGIN 3.168.192.in-addr.arpa.
2                      PTR     babuino.gonzalonazareno.org.
$TTL 86400      ; 1 day
211                    PTR     nas.gonzalonazareno.org.
```

Editamos el fichero /var/cache/bind/db.gonzalonazareno con los datos actuales dejándolo así:

```
papion:/var/cache/bind# nano db.gonzalonazareno
```

```
$ORIGIN .
$TTL 86400      ; 1 day
gonzalonazareno.org     IN SOA  papion.gonzalonazareno.org. Postmast$
34734      ; serial
21600      ; refresh (6 hours)
3600       ; retry (1 hour)
604800     ; expire (1 week)
21600      ; minimum (6 hours)
)
NS      papion.gonzalonazareno.org.
MX      10 mail.gonzalonazareno.org.
$ORIGIN gonzalonazareno.org.
aduana                  A       192.168.1.1
correo                  A       212.36.75.14
informatica             CNAME   lavadora

$TTL 86400      ; 1 day
lavadora                A       192.168.0.2
```

```
macaco                  A       192.168.0.10
papion                  A       192.168.2.2
mandril                 A       192.168.2.3
babuino                 A       192.168.3.2
mail                    A       212.36.75.14
nas                     A       192.168.1.211
$TTL 86400      ; 1 day
router                  A       192.168.0.1
smtp                    CNAME   lavadora
www                     A       212.36.74.94
```

Editamos el fichero named.conf.local editando las ultimas lineas así:

```
include "/etc/bind/rndc.key";

controls {
        inet 192.168.2.2 port 953
        allow { 192.168.2.1; } keys { "rndc-key"; };
        inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

Sin estas directivas allow solo se nos permiten consultas al servidor dns dentro de su misma red local, con este parámetro podremos hacer consultas externas. Ponemos la ip de macaco que es el nuevo servidor dhcp que nos actualizará el dns.

Probamos a arrancar el servicio:

```
papion# /etc/init.d/bind9 start
```

No nos arranca: Failed.
Miramos el log y vemos lo siguiente:

```
     Apr 28 19:52:37 papion named[5426]: /etc/bind/named.conf.local:16: zone
'168.192.in-addr.arpa': already exists previous      definition:
/etc/bind/zones.rfc1918:20
     Apr 28 19:52:37 papion named[5426]: loading configuration: failure
     Apr 28 19:52:37 papion named[5426]: exiting (due to fatal error
```

Por lo tanto vemos que se debe a un error en nuestro zones.rfc1918 que no esta como el de aduana que se nos olvidó importarlo, por lo tanto lo importamos también:

```
     papion# scp jesuslucas@192.168.0.1:/etc/bind/zones.rfc1918 /etc/bind/
```

Probamos ahora a arrancar bind:

```
     papion:/etc/bind# /etc/init.d/bind9 start
     Starting domain name service...: bind9.
```

•Comprobando el Dns y Host:

HOST:

```
papion:~# hostname -f
.f
```

ERROR

```
papion:~# hostname
Unknow server error
```

ERROR

Lo solucionamos:

```
papion:~# hostname papion
papion:~# hostname
papion
papion:~# hostname -f
papion.gonzalonazareno.org
```

DNS:

Editamos el fichero /etc/resolv.conf

```
domain gonzalonazareno.org
search gonzalonazareno.org
#nameserver 192.168.2.1
nameserver 127.0.0.1
```

Realizamos dig de pruebas y comprobamos un correcto funcionamiento del servicio,

Copiamos la clave del dns de papion a macaco:

```
macaco# scp root@papion:/etc/bind/rndc.key /etc/dhcp3
macaco# chown root:root /etc/dhcp3/rndc.key
```

MACACO :/etc/dhcp3/dhcpd.conf

```
################################################
# Líneas para la actualización del servidor DNS:
server-identifier       macaco;
ddns-updates            on;
ddns-domainname         "gonzalonazareno.org";
ddns-update-style       interim;
ddns-rev-domainname     "in-addr.arpa.";
#deny                    client-updates;

 include                 "/etc/dhcp3/rndc.key";

zone gonzalonazareno.org. {
      primary 192.168.2.2;
      key rndc-key;
      }
      zone 168.192.in-addr.arpa. {
      primary 192.168.2.2;
      key rndc-key;
      }
```

Configuramos un rango de pruebas para la interfaz virbr1:

```
subnet 192.168.2.0 netmask 255.255.255.0 {
            range 192.168.2.4 192.168.2.200;
            option routers 192.168.2.1;
            option domain-name "gonzalonazareno.org";
            option domain-name-servers 192.168.2.2;
            option broadcast-address 192.168.2.255;
#       option ntp-servers 192.168.2.2;
}
```

Debemos configurar para pruebas también que escuche y realice su función por la interfaz virbr1.

Para ello editamos el fichero:

/etc/default/dhcp3-server

```
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
#  installed  at  /etc/default/dhcp3-server  by  the  maintainer
scripts

#
# This is a POSIX shell fragment
#

# On what interfaces should the DHCP server (dhcpd) serve DHCP
requests?
 #        Separate multiple interfaces with spaces, e.g. "eth0
eth1".
INTERFACES="virbr1"
```

Borramos el registro de clientes por si había:

```
echo "" > /var/lib/dhcp3/dhcpd.leases~
echo "" > /var/lib/dhcp3/dhcpd.leases
```

Iniciamos el servicio dhcp: macaco# /etc/init.d/dhcp3-server start

Realizamos una prueba:

Borramos a mandril del dns de mandril quitando su entrada de resolución directa en /var/cache/bind/db.gonzalonazareno

Editamos la interfaz de red de mandril:

```
mandril:~# nano /etc/network/interfaces
auto eth0
iface eth0 inet dhcp
```

Reiniciamos el servicio dns:

```
papion#  /etc/init.d/bind9 restart
```

Reiniciamos el servicio dhcp:

```
macaco#  /etc/init.d/dhcp3-server restart
```

Realizamos un dig en macaco:

```
dig mandril.gonzalonazareno.org
```

Vemos que no hay respuesta (answer).

•Clientes DNS + DHCP.

El principal requisito que debe cumplir un cliente DHCP para funcionar en este entorno es que debe enviar el nombre del host (hostname) en la petición inicial. El cliente más habitual en las distribuciones linux es dhcp3-client, que no viene configurado inicialmente para enviar el hostname. Para solucionar esto editamos el fichero /etc/dhcp3/dhclient.conf e incluimos la línea:

```
send host-name mandril;
```

Entramos a mandril y realizamos un dhclient. Nos renueva la ip 192.168.2.4, ya que el rango establecido anteriormente empezaba en la 2.4

Probamos a realizar ahora un dig y vemos que ahora tenemos respuesta, también podemos observar que en papion que se crearon los ficheros db.gonzalonazareno.jnl y el inverso.

* Posteriormente habrá que quitar el entorno de pruebas y adaptarlo para la interfaz ETH2 ya que ahora la tenemos con la interfaz VIRBR1.

5.3 Instalando Kerberos

a) Instalando Kerberos

```
papion:~# aptitude install krb5-kdc krb5-admin-server
```

Debconfig: Servidor Kerberos y servidor administrativo: papion.gonzalonazareno.org

Editamos archivo de configuración: /etc/krb5kdc/kdc.conf

```
[kdcdefaults]
     kdc_ports = 88

     [realms]
     GONZALONAZARENO.ORG = {
           database_name = /var/lib/krb5kdc/principal
           admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
           acl_file = /etc/krb5kdc/kadm5.acl
           key_stash_file = /etc/krb5kdc/stash
           kdc_ports = 88
           max_life = 10h 0m 0s
           max_renewable_life = 7d 0h 0m 0s
           master_key_type = des3-hmac-sha1
           supported_enctypes = aes256-cts:normal arcfour-hmac:normal des3-
hmac-sh$
           default_principal_flags = +preauth
         }
```

Para inhabilitar por completo la utilización de Kerberos:

Editamos : /etc/default/krb5-kdc

```
 KRB4_MODE=disable
RUN_KRB524D=false
```

- Configuración del cliente Kerveros de Papion

Editamos : /etc/krb5.conf

```
 [libdefaults]
                default_realm = GONZALONAZARENO.ORG
          [realms]
                GONZALONAZARENO.ORG = {
                      kdc = papion.gonzalonazareno.org
                      admin_server = papion.gonzalonazareno.org
                }
          [domain_realm]
                .gonzalonazareno.org = GONZALONAZARENO.ORG
                gonzalonazareno.org = GONZALONAZARENO.ORG
```

- Puesta en marcha de los servicios

Para definir el realm de nuestro servidor Kerberos, ejecutaremos la siguiente instrucción que se incluye en Debian:

```
     papion :# krb5_newrealm
```

Tiene de contraseña la misma que root de las máquinas.

Iniciamos los servicios:

```
     papion :# /etc/init.d/krb5-kdc start
     papion :# /etc/init.d/krb5-admin-server start
```

Comprobamos que se han iniciado correctamente:

```
papion:~# netstat -putan |grep "kadmind"
```

```
cp          0       0 0.0.0.0:749               0.0.0.0:*                   LISTEN
7574/kadmind
  udp               0           0  0.0.0.0:464                          0.0.0.0:*
7574/kadmind
```

```
papion:~# netstat -putan |grep "krb5kdc"
```

```
udp 0 0 192.168.2.2:88    0.0.0.0:*        570/krb5kdc
udp6           0           0           fe80::216:36ff:fe43::88           :::*
7570/krb5kdc
```

Kdc escucha en el puerto 88/udp, mientras que kadmind escucha en los puertos 749/tcp para utilizar la aplicación kadmin y 464/udp para kpasswd.
Ahora es posible realizar una conexión local con el servidor kadmin donde podemos crear o modificar los principales:

```
papion:~# kadmin.local
Authenticating as principal root/admin@GONZALONAZARENO.ORG with password.
```

Vamos a comprobar los principales que se generan de forma automática al instalar el servidor:

```
      kadmin:
            kadmin.local:  list_principals
            K/M@GONZALONAZARENO.ORG
            kadmin/admin@GONZALONAZARENO.ORG
            kadmin/changepw@GONZALONAZARENO.ORG
            kadmin/history@GONZALONAZARENO.ORG
            kadmin/papion.gonzalonazareno.org@GONZALONAZARENO.ORG
            krbtgt/GONZALONAZARENO.ORG@GONZALONAZARENO.ORG
```

Vamos a crear principales para los equipos papion, macaco, babuino y mandril y el servicio ldap de papion, estos se generarán con una clave aleatoria:

```
            kadmin.local: add_principal -randkey host/papion.gonzalonazareno.org
            kadmin.local:                 add_principal                 -randkey
host/babuino.gonzalonazareno.org
            kadmin.local: add_principal -randkey host/macaco.gonzalonazareno.org
            kadmin.local:                 add_principal                 -randkey
host/mandril.gonzalonazareno.org
            kadmin.local: add_principal -randkey ldap/papion.gonzalonazareno.org
```

b) Clientes del directorio Kerberos + LDAP.

1- Configurar para que hostname -f veamos el FQDN : host.gonzalonazareno.org
Para ello editamos el fichero hosts.

```
      127.0.1.1 host.gonzalonazareno.org host
```

Si nos diera hostname -f Unknow server error:
Editar también /etc/hostame y colocar el nombre de la máquina
También realizar #hostanme host y comprobar ahora hostname -f

Lo realizamos en mandril Es dedir que copies lo que se arai al acerlo en madril.

2- Instalar libnss-ldap, configurar para que conecte con el LDAP de papion y configuración nsswitch para la resolución de nombres de usuario y grupos:

```
 mandril:# apt-get install --no-install-recommends libnss-ldap
```

No recommends para que no instale libpam-ldap, aunque lo tendremos que instalar más adelante.

Del servicio de autenticación debe encargarse Kerberos y configurar los siguientes puntos:

```
 mandril:#dpkg-reconfigure libnss-ldap
```

◦ Identificador del servidor LDAP: ldap://papion.gonzalonazareno.org
◦ Nombre distinguido (dn) de la base: dc=gonzalonazareno,dc=org
◦ Versión de LDAP: 3
◦ Cuenta del administrador: (ignorar)
◦ Contraseña del administrador: (ignorar)

Configuraremos también el cliente ldap:

```
mandril :#nano /etc/ldap/ldap.conf

BASE    dc=gonzalonazareno,dc=org
URI     ldap://papion.gonzalonzareno.org
```

Modificación de /etc/nsswitch.conf:

Editamos este fichero y modificamos las líneas correspondientes a passwd y group:

```
passwd:         compat ldap
group:          compat ldap
```

•Comprobación de funcionamiento:

Mediante la utilización de getent:

```
papion :# getent passwd
papion :# getent group

mandril:~# getent passwd mmuller
mmuller:*:2253:2001:Mariló Müller:/home/mmuller:
mandril:~# getent group profesores
profesores:*:2000:alberto.molina,josedom,raul,jesus.moreno,mjesus,jtagua,raquel
,conchi,mftienda
```

3- Instalar y configurar el sevidor de hora ntp para que obtenga la hora del servidor de hora de papion.

```
mandril:~# aptitude install ntp
```

Configuramos : /etc/ntp.conf para que use el servidor de hora de papion.

```
mandril:~# nano /etc/ntp.conf
           # You do need to talk to an NTP server or two (or three).
           server papion.gonzalonazareno.org
           # pool.ntp.org maps to about 1000 low-stratum NTP servers.   Your
server will
           # pick a different set every time it starts up.   Please consider
```

```
joining the
            # pool: <http://www.pool.ntp.org/join.html>
            #server 0.debian.pool.ntp.org iburst dynamic
            #server 1.debian.pool.ntp.org iburst dynamic
            #server 2.debian.pool.ntp.org iburst dynamic
            #server 3.debian.pool.ntp.org iburst dynamic
```

```
mandril:~# /etc/init.d/ntp restart
      Stopping NTP server: ntpd.
      Starting NTP server: ntpd.
```

Tras iniciar el demonio, podemos ver que esta sincronizado con el comando :

```
mandril:~# ntpq -np
 remote           refid      st t when poll reach   delay   offset  jitter
 192.168.2.2     .INIT.          16 u   14   64    2    0.671   -7.868   0.001
```

4- Instalación del cliente Kerberos:

```
mandril:~# aptitude install krb5-config krb5-user
```

Modificamos /etc/krb5.conf igual que en papion.

```
            [libdefaults]
                  default_realm = GONZALONAZARENO.ORG
            [realms]
                  GONZALONAZARENO.ORG = {
                        kdc = papion.gonzalonazareno.org
                        admin_server = papion.gonzalonazareno.org
                  }
            [domain_realm]
                  .gonzalonazareno.org = GONZALONAZARENO.ORG
                  gonzalonazareno.org = GONZALONAZARENO.ORG
```

Vamos a probar el funcionamiento añadiendo un principal de prueba llamado "prueba" con clave "prueba".

```
papion:~# kadmin.local
```

```
      Authenticating as principal root/admin@GONZALONAZARENO.ORG with password.
      kadmin.local:  add_principal prueba
      WARNING: no policy specified for prueba@GONZALONAZARENO.ORG; defaulting to
no policy
      Enter password for principal "prueba@GONZALONAZARENO.ORG":
      Re-enter password for principal "prueba@GONZALONAZARENO.ORG":
      Principal "prueba@GONZALONAZARENO.ORG" created.
```

Cuando la prueba sea satisfactoria lo borraremos.
Prueba de funcionamiento:

Esta instrucción muestra los tickets de la sesión de usuario, si se ejecuta antes de autenticarse se

obtiene esta salida (el parámetro -5 es para que sólo utilice Kerberos5):

```
mandril:~# klist -5
```

```
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

Realizamos un kinit y nos autenticamos como "prueba":

```
mandril:~# kinit prueba
```

```
Password for prueba@GONZALONAZARENO.ORG:
```

```
mandril:~# klist -5
```

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: prueba@GONZALONAZARENO.ORG
Valid starting     Expires            Service principal
05/03/10 20:15:49  05/04/10 06:15:49
 krbtgt/GONZALONAZARENO.ORG@GONZALONAZARENO.ORG
renew until 05/04/10 20:15:46
```

La prueba realizada es satisfactoria como podemos observar.

Borramos los restos de la pruebas en mandril:

```
mandril:~# kdestroy
```

```
mandril:~# klist -5
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_0)
```

Borramos los restos de la pruebas en papion:

```
kadmin.local:  delete_principal prueba
Are you sure you want to delete the principal "prueba@GONZALONAZARENO.ORG"?
(yes/no): yes
Principal "prueba@GONZALONAZARENO.ORG" deleted.
Make sure that you have removed this principal from all ACLs before reusing.
```

5- Configuración SASL/GSSAPI

```
mandril:~#aptitude install libsasl2-modules-gssapi-mit
```

6- Configuración del PAM:

Antes de nada, como vamos a tocar una zona sensible de nuestro sistema, realizaremos una copia de seguridad del PAM:

```
mandril:~# cp -r /etc/pam.d /etc/pam.d.old
```

Para que nuestro sistema sea capaz de autenticar contra kerberos hay que instalar el paquete libpam-krb5 :

```
mandril:~# aptitude install libpam-krb5
```

- Servidores Kerberos para su dominio: papion.gonzalonazareno.org
- Servidor administrativo para su dominio Kerberos: papion.gonzalonazareno.org

Para que poder utilizar nuestro servidor LDAP en el Pam deberemos instalar la librería correspondiente.

```
mandril:~# aptitude install libpam-ldap
```

Configuración:
- No administrador local
- Usuario para ldap: gusuario-nss

Los ficheros common-* del directorio /etc/pam.d/ quedarán exactamente como los de lavadora por ello los importamos de esta.

A continuación pasamos a realizar algunas *pruebas de funcionamiento*:

- Tengo mi usuario en Ldap : Luks89, con mi contraseña.

- Creo en papion mi principal para kerberos:

```
kadmin.local add_principal luks89
```

- Pongo mi misma contraseña que la da LDAP.

- Voi a Mandril y hago:

```
mandril:/etc/pam.d# login luks89
Password:
Linux mandril 2.6.26-2-amd64 #1 SMP Thu Nov 5 02:23:12 UTC 2009 x86_64
The programs included with the Debian GNU/Linux system are free software; the
exact distribution terms for each program are described in the  individual
files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
     Creando directorio '/home/luks89'.
     luks89@mandril:~$ ls -l
     total 0
     luks89@mandril:~$ touch hola
     luks89@mandril:~$ ls -l
     total 0
     -rw-r--r-- 1 luks89 alumnos 0 may  3 21:14 hola
```

Podemos ver que nos deja loguearnos, nos crea nuestro home al no existir este (esta configuración se encuentra en los ficheros del pam que importamos de lavadora), y podemos observar como funciona perfectamente la resolución de nombre de usuario y grupos.

Eliminamos restos de las pruebas:

```
papion:~# kadmin.local delprinc luks89
```

5.4 Migración de los servicios de la DMZ

a) Migrando el Servidor Web Apache

```
babuino:~# aptitude install apache2
```

En este caso el sitio web no va a radicar en el directorio /var/www como es habitual en Debian sino en /srv/www que es su sitio natural y más lógico.

Importaremos los sitios de apache desde lavadora:

```
babuino:/etc/apache2/sites-enabled# scp jesuslucas@lavadora:/etc/apache2/sites-available/* ../sites-available/
```

Activamos el sitio SSL :

```
 babuino:/etc/apache2# a2ensite default-ssl
Enabling site default-ssl.
```

•Ajustes de configuración

Se edita el fichero /etc/apache/mods-enabled/status.conf y se añade la línea referente a papion para que se pueda comprobar el estado del servidor desde la red local:

```
 <Location /server-status>
 SetHandler server-status
 Order deny,allow
 Deny from all
 Allow from papion.gonzalonazareno.org
           Allow from localhost ip6-localhost
     #    Allow from .example.com
     </Location>
```

Se edita el fichero /etc/apache2/mods-enabled/negotiation.conf y se modifica la directiva LanguagePriority poniendo en primer lugar español:

```
     LanguagePriority es en ca cs da de el eo et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv tr zh-CN zh-TW
```

Podemos ver en default-ssl los certificados que estamos usando :

```
     SSLCertificateFile /etc/ssl/private/apache.pem
     SSLCertificateKeyFile /etc/ssl/private/private.key
     SSLCertificateChainFile /etc/ssl/certs/cacert.org.pem
```

Por lo tanto deberemos importarlos de lavadora a babuino con ssh:

```
babuino:~# scp lavadora:/etc/ssl/private/apache.pem /etc/ssl/private/
babuino:~# scp lavadora:/etc/ssl/private/private.key /etc/ssl/private/
babuino:~# scp lavadora:/etc/ssl/certs/cacert.org.pem /etc/ssl/certs/
```

Nos aseguramos que los archivos de la carpeta private tengan de propietario root y permisos 644.

Reiniciamos el servidor apache:

```
babuino:~# /etc/init.d/apache2 restart
Starting web server: apache2apache2: apr_sockaddr_info_get() failed for babuino
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
```

El siguiente error nos ocurre debido a una mala configuración del fichero /etc/hosts , por lo tanto lo editaremos y corregiremos de la siguiente manera:

```
babuino:~# nano /etc/hosts
```

```
127.0.0.1 localhost
127.0.1.1 babuino.gonzalonazareno.org babuino
# Las siguientes lÃneas son recomendables en equipos que pueden
# utilizar IPv6
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Instalamos lynx para poder realizar pruebas de funcionamiento sin disponer de entorno gráfico.

```
Babuino:~# aptitude install lynx
babuino:~# nano /srv/www/index.html
babuino:~# lynx localhost
```

b) Migrando las aplicaciones web.

Para migrar todas las aplicaciones web que están en Lavadora sera imprescindible que se mantengan los permisos y propietarios que los archivos de estas poseen, por ello no deberemos usar el comando scp para importarlas. Por lo tanto para que este requisito se cumpla lo que haremos ser comprimir el directorio donde se encuentran nuestras aplicaciones con tar y posterior mente pasar este comprimido a babuino mediante scp.

```
jesuslucas@lavadora:~$ tar -czvf /tmp/web.tar.gz /www
babuino:~# scp -r jesuslucas@lavadora:/tmp/web.tar.gz /srv
babuino:~# tar -zxvf /tmo/web.tar.gx -C /srv
```

Para la aplicación moodle sera necesario el directorio moodledata que lo deberemos crear y darle todos los permisos.

```
babuino:~# mkdir /srv/moodledata
babuino:~# chmod 777 /srv/moodledata/
```

Una vez tenemos todos los archivos de nuestras aplicaciones en Babuino procederemos a instalar el sistema gestor de bases de datos que usan, en este caso MySQL 5, que será instalado en Papion como pudimos ver en el esquema de la organización de la nueva red.

Posteriormente configuraremos de forma adecuada cada aplicación para que funcione de forma correcta.

c) Migrando el servidor de Bases de Datos MySQL 5.

Instalamos en Papion el servidor de bases de datos MysqlServer 5 .

```
papion:~# aptitude install mysql-server
```

Debconf nos pregunta la contraseña de root que va a ser la misma de root del mysql de lavadora.

Luego cargaremos una copia de seguridad de la base de datos.

```
papion:~# scp jesuslucas@lavadora:/srv/copia_bd.sql .
```

Deberíamos copiar este archivo también para que no se viera afectado el suuario debian-sys-maint:

```
papion:~# scp jesuslucas@lavadora:/etc/mysql/debian.conf
```

Pero como no tenemos permisos sobre este luego tendremos que modificar el usuario debians-sys-maint de nuestra copia de seguridad colocándole la contraseña que esta definida en el archivo debia.conf actual.

Ahora cargaremos la copia de seguridad en MySQL.

```
papion:~# mysql -p < copia_bd.sql
```

```
papion:~# mysql -p
```

```
mysql> show databases;

| Database           |
+--------------------+
| information_schema |
| drupal             |
| empresas           |
| moodle             |
| mysql              |
| wikidb             |
+--------------------+
6 rows in set (0.00 sec)
```

Podemos ver que tenemos todas las bases de datos importadas.

Al realizar la importación de esta manera también importamos el diccionario de datos de MySQL, que es la base de datos information_schema, en esta están definidos los usuarios entre otras muchos detalles. Esto nos origina un problema, MySQL dispone del usuario debian-sys-maint que es un usuario con privilegios de root el cual dispone mysql para tareas administrativas internas, pero su contraseña esta definida en el archivo debian.cnf a la vez que en la base de datos information_schema y ahora mismo no coinciden por lo que nos dará muchos problemas cualquier inicio o reinicio de la MySQL.

Para arreglar esto solo tendremos que entrar como root a la base de datos information_schema y cambiar el password de este usuario mediante una sentencia SQL simple por el que está establecido

en el archivo debian.cnf.

```
papion:~# cat /etc/mysql/debian.cnf
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host     = localhost
user     = debian-sys-maint
password = sgJG6vzd94gwEk9C
socket   = /var/run/mysqld/mysqld.sock
[mysql_upgrade]
user     = debian-sys-maint
password = sgJG6vzd94gwEk9C
socket   = /var/run/mysqld/mysqld.sock
basedir  = /usr
```

```
papion:~#  mysql -p
mysql > GRANT ALL PRIVILEGES ON *.* TO 'debian-sysmaint'@'localhost' IDENTIFIED
BY 'sgJG6vzd94gwEk9C' WITH GRANT OPTION;
```

Una vez realizado ya tenemos solucionado el problema.

Otro de los problemas que nos encontramos es que por defecto MySQL viene configurado para aceptar solo conexiones provenientes desde localhost, y esto no es lo que queremos puesto que nuestras aplicaciones se encuentran en una red distinta a la de papion.

Para solucionar este problema deberemos establecer el parámetro bindaddress del archivo de configuración de MySQL a 0.0.0.0 .

```
papion:~# nano  /etc/mysql/my.conf
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
Bind-address            = 0.0.0.0
```

```
papion:~# mysql -p
 Enter password:
 Welcome to the MySQL monitor.  Commands end with ; or \g.
 Your MySQL connection id is 30
 Server version: 5.0.51a-24+lenny3 (Debian)

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



mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'babuino.gonzalonazareno.org'
IDENTIFIED BY 'XXXXX';
```

Donde XXXXX es la contraseña que tenia el root del servidor MySQL de lavadora.

```
babuino:~# aptitude install phpmyadmin
babuino:/usr/share/phpmyadmin# mv config.inc.php config.inc.php.original
babuino:/usr/share/phpmyadmin# nano config.inc.php
*/
           $cfg['blowfish_secret'] = 'secreto'; /* YOU MUST FILL IN THIS FOR
```

```
COOKIE AUTH! */

            /*
             * Servers configuration
             */
            $i = 0;
            /*
             * First server
             */
            $i++;
            /* Authentication type */
            $cfg['Servers'][$i]['auth_type'] = 'cookie';
            /* Server parameters */
            $cfg['Servers'][$i]['host'] = 'papion.gonzalonazareno.org';
```

```
babuino:~# aptitude install mysql-client
babuino:~# mysql -u root -p -h 192.168.2.2
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 31
Server version: 5.0.51a-24+lenny3 (Debian)

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

mysql> quit
```

d) Instalando PHP5

```
babuino:~# aptitude install php5 php5-mysql php5-ldap php5-gd php5-curl php5-
xmlrpc
```

Esto provoca que se instale apache2-mpm-prefork y se elimine apache2-mpm-worker.
Si ahora intentamos instalar worker:

```
 Los siguientes paquetes estÃ¡n ROTOS:
 libapache2-mod-php5
 Se instalarÃ¡n los siguiente paquetes NUEVOS:
 apache2-mpm-worker
 Se ELIMINARÃN los siguientes paquetes:
 apache2-mpm-prefork{a}
  0 paquetes actualizados, 1 nuevos instalados, 1 para eliminar y 0 sin
actualizar.
 Necesito descargar 0B/258kB de ficheros. DespuÃ©s de desempaquetar se usarÃ¡n
8192B.
 No se satisfacen las dependencias de los siguientes paquetes:
  libapache2-mod-php5: Depende: apache2-mpm-prefork (> 2.0.52) pero no es
instalable o
 apache2-mpm-itk pero no es instalable
 Las acciones siguientes resolverÃ¡n estas dependencias

 Eliminar los paquetes siguientes:
 libapache2-mod-php5
```

```
Instalar los paquetes siguientes:
php5-cgi [5.2.6.dfsg.1-1+lenny8 (stable)]

                La puntuaciÃ³n es -250
```

Por ahora dejaremos que apache trabaje en modo prefork en el futuro se verá la opción de pasarlo a worker para mejorar el rendimiento.

e) Activación del Módulo SSL de Apache:

Al instalar los paquetes anteriores por dependencia se instala el paquete ca-certificates que nos haría falta en el siguiente paso:

```
babuino:/srv/www/apuntes# a2enmod ssl
     Enabling module ssl.
     See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure
SSL and create self-signed certificates.
     Run '/etc/init.d/apache2 restart' to activate new configuration!
```

Ya disponíamos anteriormente de nuestro certificado ca-cert que importamos desde lavadora por lo tanto no tenemos que realizar nadas más para activar el modulo ssl de apache.

```
           babuino:/srv/www/apuntes# /etc/init.d/apache2 restart
           Restarting web server: apache2 ... waiting .
```

f) Migrando el Servidor de Correo Postfix

Algunas de las aplicaciones web utilizan un servidor de correo, fundamentalmente para enviar notificaciones a los usuarios, por lo que es adecuado instalar un servidor de correo en este equipo.

Puesto que ya existe un servidor de correo del dominio gonzalonazareno.org en un equipo externo, los registros MX del DNS apuntan a él, por lo que babuino se encargará solo de gestionar el correo del subdominio informatica.gonzalonazareno.org.

Como MTA se va a utilizar postfix, por tanto habrá que los tendremos que instalar como primer paso:

```
babuino:~# aptitude install postfix
```

Los parámetros que hay que incluir durante la instalación son:
Sitio de Internet

```
System mail name: informatica.gonzalonazareno.org
```

En cualquier caso, tras la instalación es mejor editar el fichero /etc/postfix/main.cf y comprobar o modificar las siguientes líneas:

```
myhostname = babuino.gonzalonazareno.org
…
           myorigin = /etc/mailname
           mydestination = $myhostname, $myorigin, localhost
```

```
            relayhost =
            mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

Además, el fichero /etc/mailname debe tener el siguiente contenido:

```
      informatica.gonzalonazareno.org
```

De esa manera este equipo recibirá correo para los destinos adecuados y enviará correo poniendo como remitente @informatica.gonzalonazareno.org

```
      babuino:~# nano /etc/mailname
      babuino:~# nano /etc/postfix/main.cf
      babuino:~# /etc/init.d/postfix restart
      Stopping Postfix Mail Transport Agent: postfix.
      Starting Postfix Mail Transport Agent: postfix.
```

•Alias de Correo

Editamos el fichero /etc/aliases y añadimos los alias :

```
      root: alberto, jose, jesus
```

y ejecutamos:

```
      babuino:~# newaliases
```

g) Adaptando las aplicaciones web a la nueva configuración.

Disponemos del puerto 1080 redireccionado a babuino para probar las aplicaciones mientras realizamos la migración, estando completamente operativo todavía todas las aplicaciones de lavadora por el puerto 80.

- Plataforma Moodle.

Editamos el fichero: /srv/www/plataforma/config.php y colocamos los usuarios y parámetros de la base de datos de papion correctamente.

```
 <?php  /// Moodle Configuration File

 unset($CFG);

 $CFG->dbtype    = 'mysql';
 $CFG->dbhost    = 'papion.gonzalonazareno.org';
 $CFG->dbname    = 'moodle';
 $CFG->dbuser    = 'root';
 $CFG->dbpass    = 'XXXXX';
            $CFG->dbpersist =  false;
            $CFG->prefix    = 'mdl_';

            $CFG->wwwroot                                                   =
'http://informatica.gonzalonazareno.org:1080/plataforma';
```

```
            $CFG->dirroot   = '/srv/www/plataforma';
            $CFG->dataroot  = '/srv/moodledata';
            $CFG->admin     = 'admin';

            $CFG->directorypermissions = 00777;   // try 02777 on a server in
Safe Mode

            require_once("$CFG->dirroot/lib/setup.php");
// MAKE SURE WHEN YOU EDIT THIS FILE THAT THERE ARE NO SPACES, BLANK LINES,
// RETURNS, OR ANYTHING ELSE AFTER THE TWO CHARACTERS ON THE NEXT LINE.
            ?>
```

Nos saldrá el mensaje siguiente al entrar en:
http://informatica.gonzalonazareno.org:1080/plataforma

```
Error: Database connection failed.
  It is possible that the database is overloaded or otherwise not running
properly.

The site administrator should also check that the database details have been
correctly specified in config.php

Esto es porque el usuario root no tiene privilegios de conexion fuera de
localhost (  papion ) .
```

Para solucionar este error debemos darle privilegio de conexión desde 192.168.2.1 (interfaz de macaco que enruta la petición a papion 192.168.2.2) al usuario root.

```
Papion:~# mysql -p
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'XXXXXX';
```

Hay Que cambiarlo esto más tarde para configurarlo con los host permitidos y no para todos los host (%).

También debemos editar el fichero index.php con la dirección de prueba:

```
<?
header("Location:http://informatica.gonzalonazareno.org:1080/plataforma");
?>
```

Funciona pero por ahora alguna pruebas no las podemos realizar debido a que no podemos probar SSL login por la configuración del firewall de aduana.

Una vez visto que funciona la aplicación, excepto el login que se solucionará una vez migremos físicamente todo, debemos copiar el directorio moodledata donde se encuentran todos lo ficheros y archivos subidos por los usuarios entre otros datos.

Copiamos el directorio de datos de moodle: moodledata:

```
babuino:/srv# scp -r jesuslucas@lavadora:/srv/moodledata .
babuino:/srv# chown -R www-data:www-data moodledata/
```

Con esto ya tenemos la aplicación moodle funcionando correctamente.

- Aplicación MediaWiki

Editaremos el archivo de configuración siguiente y colocaremos los parámetros correctos correspondientes a la actual configuración del servidor MySQL y LDAP:

wiki/config/LocalSettings.php

```
## Database settings
$wgDBtype           = "mysql";
$wgDBserver         = "papion.gonzalonazareno.org";
$wgDBname           = "wikidb";
$wgDBuser           = "wikiuser";

#...

# ValidaciÃ³n LDAP
require_once( "$IP/extensions/LdapAuthentication.php" );
$wgAuth = new LdapAuthenticationPlugin();
$wgLDAPDomainNames = array("GONZALONAZARENO.ORG");
$wgLDAPServerNames =array("GONZALONAZARENO.ORG"=>"papion.gonzalonazareno.org");
$wgLDAPUseLocal = true;
$wgLDAPEncryptionType = array("GONZALONAZARENO.ORG"=>"clear");
$wgLDAPBaseDNs = array("GONZALONAZARENO.ORG"=>"dc=gonzalonazareno,dc=org");
$wgLDAPSearchAttributes = array("GONZALONAZARENO.ORG"=>"uid");
$wgLDAPGroupsPrevail = array("GONZALONAZARENO.ORG"=>true);
$wgLDAPGroupNameAttribute = array("GONZALONAZARENO.ORG"=>"cn");
```

La aplicación MediaWiki también guarda información del servidor ldap que se está utilizando en su base de datos, por lo que debemos actualizar el registro en el que se encuentra dicha información.

```
Papion:~# mysql -p
```

```
mysql> use wikidb;
mysql>     UPDATE     `moodle`.`mdl_config_plugins`     SET     `value`     =
'papion.gonzalonazareno.org' WHERE `mdl_config_plugins`.`id` =24 LIMIT 1 ;
```

Así la tenemos ya funcional la aplicación excepto el login por ssl por el mismo motivo que no funcionaba en la plataforma moodle.

- Aplicación Empresa

Modificamos el archivo de configuración en este caso /srv/www/empresa/funciones.php y le colocamos al configuración correspondiente, donde XXXXX es la contraseña de root de mysql.

```
     //Conecta a la base de datos.
     $bd=@mysql_connect(papion.gonzalonazareno.org,"root","XXXXX")          or
die("Error en la conexion a la base de datos");
```

Esta aplicación no funciona ya que esta incompleta, nos sale el siguiente error al intentar acceder:

```
Parse error:syntax error, unexpected T_DNUMBER in
/srv/www/empresa/funciones.php on line 95
```

- Aplicación Drupal : Portal

Buscamos los archivos que hay que modificar realizando un grep recursivo filtrando por la contraseña de root del usuario de mysql que disponía lavadora (XXXXX).

```
babuino:/srv/www/portal# grep -r XXXXX .
./sites/default/settings.php:$db_url = 'mysqli://root:XXXXX@localhost/drupal';
```

Modificado:

```
$db_url = 'mysqli://root:XXXXX@papion/drupal';
```

También debemos modificar las entradas donde tengamos información del servidor LDAP en la base de datos de esta aplicación. Para ello usaremos la siguiente sentencia sql:

```
Papion:~# mysql -p
mysql> use wikidb;
mysql>UPDATE `drupal`.`ldapauth` SET `name` = 'papion', `server` =
'papion.gonzalonazareno.org' WHERE CONVERT( `ldapauth`.`name` USING utf8 ) =
'lavadora' LIMIT 1 ;
```

Probamos que funciona entrando con nuestro usuario de LDAP.

- Aplicación Gestiona

Como anteriormente no disponíamos de permisos para importarla a la vez que el resto de aplicaciones lo tendremos que realizar ahora una vez se nos han concedido dichos permisos. Usaremos la misma táctica que antes para mantener permisos y propietarios.

```
jesuslucas@lavadora:~#tar -czvf /home/jesuslucas/gestiona.tar.gz
/srv/www/gestiona --same-permissions –same-owner

babuino:~# scp jesuslucas@lavadora:/home/jesuslucas/gestiona.tar.gz .
babuino:~# tar -xvf gestiona.tar.gz -C /srv/www/
```

Editamos el archivo de configuración de la aplicación Gestiona y le colocamos los parámetros correspondientes al servidor LDAP que es papion:

```
babuino:/srv/www/gestiona/include# nano config.php
```

```
            //IP del servidor LDAP
            $server_ldap="192.168.2.2";
```

- Aplicación Awstats

Instalamos la aplicación por medio de la herramienta de paquetes aptitude:

```
babuino:/# aptitude install awstats
```

Una vez instalada importamos la configuración que dispone esta aplicación en lavadora:

```
babuino:/# scp jesuslucas@lavadora:/etc/awstats/awstats.conf /etc/awstats
```

También debemos configurar los grupos del log de apache para que sean www-data y la aplicación Awstats pueda leerlos correctamente.

```
babuino:/# chgrp -R www-data /var/log/apache2/
```

Y además editamos el fichero /etc/logrotate.d/apache2 y modificamos la linea o la añadimos si no existe:

```
create 640 root www-data
```

Una vez realizado esto ya tenemos la aplicación funcional pero no queremos perder las estadísticas que manteníamos desde hace meses en lavadora por lo que las tendremos que importar:

```
babuino:/etc/awstats#        scp        jesuslucas@lavadora:/var/lib/awstats/*\
/var/lib/awstats/
```

Y le ponemos los permisos y propietarios correctamente:

```
chmod 750 -R /var/lib/awstats
chown www-data:www-data /var/lib/awstats
```

5.5 Migración de otros servicios

a) Migrando SQUID3

Para los datos de la caché de Squid crearemos un volumen lógico para mantenerlo aislado del resto del sistema y poder extender su tamaño en caso de que esto fuera necesario.

Se realizan los siguientes pasos:

- Creamos el volumen lógico y lo formateamos como EXT3:

```
macaco:~# lvcreate -L 10G -n squid vg
macaco:~# mkfs.ext3 /dev/mapper/vg-squid
```

- Añadimos y definimos el nuevo volumen lógico en la máquina virtual Papion.

```
macaco:~# virsh destroy papion
macaco:~# nano /etc/libvirt/qemu/papion.xml
```

```
<disk type='block' device='disk'>
     <source dev='/dev/mapper/vg-squid'/>
     <target dev='vdc' bus='virtio'/>
   </disk>
```

```
macaco:~# virsh list
macaco:~# virsh define /etc/libvirt/qemu/papion.xml
macaco:~# virsh start papion
```

- A continuación lo añadimos al fstab de Papion para que se monte automáticamente:

```
papion:~# cat /etc/fstab
```

```
# file system    mount point   type    options                 dump pass
/dev/vda         /             ext3    defaults                0    1
/dev/vdc         /var/spool/squid3  ext3    defaults            0    1
/dev/vdb         none          swap    sw                      0    0
proc             /proc         proc    defaults                0    0
```

- Instalamos Squid3:

```
papion:~# aptitude install squid3
```

Editamos el fichero de configuración y comentamos las líneas:

```
#acl localnet src 10.0.0.0/8          # RFC1918 possible internal network
#acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
```

y descomentamos:

```
http_access allow localnet
```

De esta manera pueden utilizar el proxy todos los equipos del segmento de red 192.168.0.0/16.

Deberemos añadir la siguiente regla al firewall iptables ,que estará en Macaco, y añadir el siguiente parámetro a la configuración de Squid para que nuestro proxy funcione de manera transparente al usuario:

squid.conf

```
http_port 3128 transparent
```

Iptables:

```
# Regla de proxy transparente
iptables -t nat -A PREROUTING -s $AULAS -d ! 192.168.0.0/16 -p tcp --dport 80 \
-j DNAT --to $PAPION:3128
```

Más tarde habrá que configurar squid según nuestros criterios, por ahora se queda con la siguiente configuración:

```
papion:~# cat /etc/squid3/squid.conf | grep -v ^$ | grep -v ^#
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl SSL_ports port 443
```

```
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localnet
http_access allow localhost
http_access deny all
icp_access deny all
htcp_access deny all
http_port 3128 transparent
hierarchy_stoplist cgi-bin ?
access_log /var/log/squid3/access.log squid
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern (cgi-bin|\?)    0       0%      0
refresh_pattern .               0       20%     4320
icp_port 3130
coredump_dir /var/spool/squid3
```

b) Migrando Approx

Para los datos de la caché de Approx crearemos un volumen lógico por los mismos motivos que anteriormente expusimos para Squid:

Se realizan los siguientes pasos:

- Creamos el volumen lógico de 20 GB y lo formateamos como EXT3:

```
macaco:~# lvcreate -L 20G -n approx vg
macaco:~# mkfs.ext3 /dev/mapper/vg-approx
```

- Añadimos y definimos el nuevo volumen lógico en la máquina virtual Papion.

```
macaco:~# virsh destroy papion
macaco:~# nano /etc/libvirt/qemu/papion.xml
```

```
 <disk type='block' device='disk'>
      <source dev='/dev/mapper/vg-approx'/>
      <target dev='vdd' bus='virtio'/>
    </disk>
```

```
macaco:~# virsh list
macaco:~# virsh define /etc/libvirt/qemu/papion.xml
macaco:~# virsh start papion
```

- A continuación lo añadimos al Fstab de Papion para que se monte automáticamente:

```
papion:~# cat /etc/fstab
# file system    mount point   type    options                 dump pass
/dev/vda         /             ext3    defaults                0    1
/dev/vdc         /var/spool/squid3  ext3    defaults           0    1
/dev/vdd         /var/cache/approx  ext3    defaults           0    1
/dev/vdb         none          swap    sw                      0    0
proc             /proc         proc    defaults                0    0
```

- Instalamos Approx:

```
#aptitude install approx
```

Importamos el fichero de configuración /etc/approx/approx.conf.

```
papion:~# scp jesuslucas@aduana:/etc/approx/approx.conf /etc/approx/
```

Importamos la cache de paquetes que disponíamos en aduana y le damos los propietarios y permisos correspondientes:

```
papion:~# scp -r jesuslucas@aduana:/var/cache/approx/ /var/cache/
papion:~# chown -R approx:approx /var/cache/approx/
papion:~# chmod -R 755 /var/cache/approx/
```

Reiniciamos el servicio Approx:

```
papion:~# /etc/init.d/approx restart
```

c) Migrando las tareas del CRON necesarias.

```
babuino:/etc# scp jesuslucas@lavadora:/etc/cron.d/drupal /etc/cron.d
babuino:/etc# scp jesuslucas@lavadora:/etc/cron.d/moodle /etc/cron.d
babuino:/etc# scp jesuslucas@lavadora:/etc/cron.d/awstats /etc/cron.d
babuino:/etc# /etc/init.d/cron restart
Restarting periodic command scheduler: crond.
```

Para que funcione correctamente la tarea de Cron de Moodle debemos instalar php5-cli en Babuino.

```
babuino:/etc# aptitude install php5-cli
```

Editamos los archivos de configuración colocando el horario que nos convenga y estableciendo los directorios y usuarios correctamente.

```
babuino:/etc/cron.d# cat awstats
10  6  *  *  *  www-data  /usr/lib/cgi-bin/awstats.pl  -config=awstats  -update
>/dev/null
```

```
babuino:/etc/cron.d# cat drupal
20,50 * * * * root cd /srv/www/portal/ && /usr/bin/php cron.php > /dev/null
```

```
babuino:/etc/cron.d# cat moodle
```

```
0,10,20,30,40,50 * * * * root /usr/bin/php /srv/www/plataforma/admin/cron.php >
/dev/null
```

5.6 Migración física

Una vez realizada la migración de los servicios es hora de desconectar los antiguos servidores y dejar el nuevo. Este paso fue realizado por el tutor de este proyecto Alberto Molina y consistió en dos fases:

- La primera fase después de tener todos los servicios y máquinas virtuales operativas se basó en la desconexión de aduana y mantener la red unos días funcionando con papion pero manteniendo a lavadora como servidor web.

- La segunda fase se centró en la desconexión de lavadora una vez comprobado que la red seguía funcionando correctamente usando papion. Durante

Durante estas dos fases hubo que modificar el dns para que apuntara correctamente las entradas a las actuales máquinas, quedando de la siguiente manera:

Para no perder las estadísticas de Awstats y los datos de las bases de datos de MySQL de lavadora, el servidor se paró durante una hora aproximadamente, tiempo en el que se volvió a volcar los datos de Awstats sobre babuino así como las bases de datos, con la consecuente edición de los parámetros de cada tabla que hicieron falta modificar por cada aplicación como se explicó anteriormente.

Por último se procedió a realizar pruebas de funcionamiento de las aplicaciones web y servicios que aloja babuino tanto desde el exterior de la red como desde el interior y finalmente a dejar a lavadora en funcionamiento durante unos días pero como servidor secundario por si en babuino surgiera algún problema.

6. Mejoras posibles.

Una vez realizada la migración, en el proceso de desarrollo de esta se han ido observando ciertas mejoras que podrían realizarse para aumentar el rendimiento o tener un sistema más completo. A continuación expondré algunas de las recomendaciones que a mi juicio considero necesarias o complementarias. Estas podrán ser utilizadas para proyecto integrado de algún alumno del siguiente curso.

- Configuración de apache en modo Worker.
- Ajuste de las reglas de Squid.
- Ajuste de las reglas de Iptables.
- Montar un sistema de centralizado de usuarios y sus archivos con Samba o nfs4 para todos los alumnos.
- Ajustar los permisos y usuarios correspondientes a las bases de datos MySQL. Deberíamos tener solo un usuario (aparte del root) por cada base de datos que tenga todos los permisos sobre esta y será este usuario el que se usara en las aplicaciones web.
- Realización de script de backup de todo el servidor Dell al completo.
- Instalación de alguna herramienta de monitorización de los servicios como Nagios o Pandora entre otros para llevar el control de los servicios.

7. Conclusiones

Con la realización del presente proyecto se ha conseguido cumplir los objetivos inicialmente propuestos y realizar la migración de dos servidores muy poco eficientes a un servidor con bastante más potencial, así como a saber analizar los defectos que la configuración de los antiguos servidores tenían y las mejoras posibles y deseables que deberían implantarse.

También se aprendió el uso de Latex a nivel básico para ayudar en la documentación del mismo, en la cual se ha participado y se continuará participando una vez entregado este proyecto.

A nivel técnico, he comprendido que la migración de servidores que están en producción no es una tarea fácil y rápida, ya que siempre surgen numerosos problemas durante el transcurso de la misma.

Durante la realización de la migración a sido importante la necesidad de informar del estado del servidor y duda que surgieran a los otros alumnos que debían implantar sus proyectos en dicho servidor.

Este proyecto no es un proyecto cerrado, sino que puede ser y debe ser continuado por profesores y alumnos que quieran tener una red y unos servidores de calidad y máxima eficiencia.

Por último comentar que no hubiera sido posible la realización del proyecto sin la ayuda del tutor del mismo Alberto Molina.

8. Referencias y Ayudas

- Documentación del servidor “Lavadora”.
- Documentación del servidor “Aduana”.
- http://docs.webfaction.com/user-guide/moving.html
- Lista de correo de Kerberos@mit.edu
- Documentación de Alberto Molina: Servicios de Directorio en Unix Kerberos + LDAP
- http://albertomolina.wordpress.com/
- Toda la formación obtenida durante los dos cursos de Administración de sistemas Informaticos en el IES Gonzalo Nazareno de Dos Hermanas.
- Ayuda del tutor y colaborador del proyecto Alberto Molina, el cual realizó las siguientes partes:
  - Instalación del sistema de red y máquinas virtuales KVM.
  - Creación de los volúmenes lógicos necesarios para approx y squid.
  - Importación de los paquetes de approx.
  - Realización del Script de Iptables como firewall en Macaco.
  - Testeado continuo del servidor.
  - Y alguna que otras tareas que seguramente me deje atrás.