

# PRÁCTICA FINAL ADMINISTRACIÓN DE SISTEMAS

Memoria



Alberto García Martín  
Lucía Q. Iturriaga Sanz

## Índice

1.	Introducción	2
2.	Descripción del sistema y sus servicios	6
2.1.	Hostname y MOTD	6
2.2.	Servidor web	6
2.3.	Usuarios	7
2.4.	Portal web	13
2.4.1.	Registro de usuarios	18
2.5.	Correo electrónico	18
2.6.	Web personal	27
2.7.	SFTP	32
2.8.	Moodle	36
2.9.	Monitorización	38
2.10.	Control de cuotas	42
2.11.	Web de status	42
2.12.	Copias de seguridad	44
2.13.	Otros requisitos	48
2.13.1.	Archivo condiciones.txt	48
2.13.2.	Carpeta apuntes	48
2.13.3.	Página de ayuda	48
2.13.4.	Prevención de fork bombs	50
2.14.	Requisitos opcionales	50
2.14.1.	Aviso login administrador	50
2.14.2.	Bloquear páginas web a los alumnos	51
2.14.3.	Software de llamadas y videollamadas	51
2.14.4.	Almacenar copias de seguridad en un servidor remoto	53
	Bibliografía	54

## 1. Introducción











En esta práctica hemos realizado la instalación y configuración de un servidor que tiene que proporcionar diversos servicios al departamento de informática de la universidad.

En todo momento cuando no se indique la decisión tomada en un paso, se asumirá que es la opción por defecto.

En primer lugar, para poder acceder al servidor desde fuera de nuestra red de forma sencilla registramos el dominio “sysadminsolutions.cf” de forma gratuita a través de Freenom. Luego añadimos los registros DNS necesarios para apuntar a nuestra IP pública y poder acceder a todos nuestros servicios.

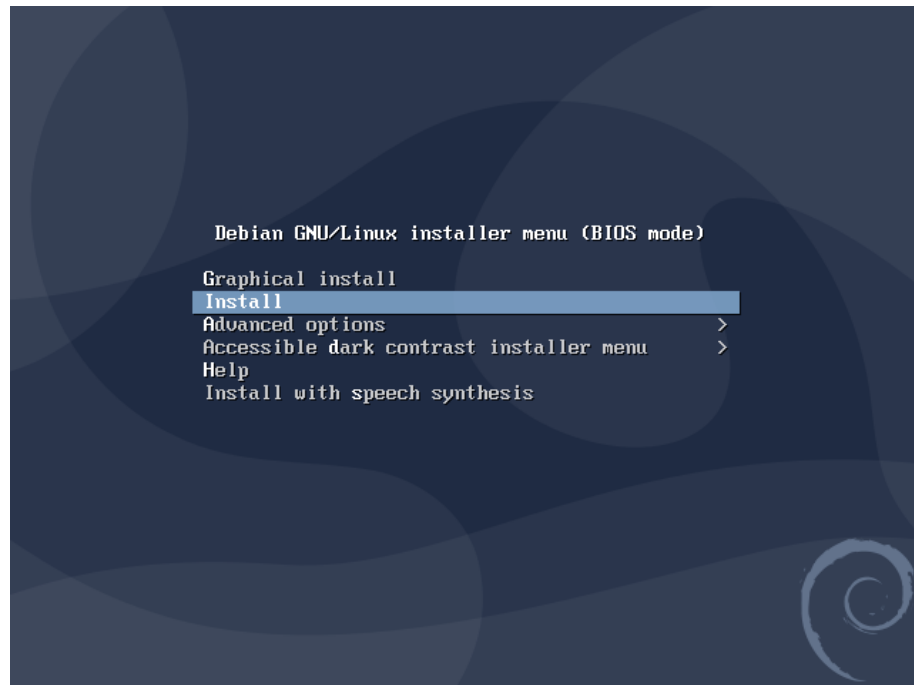
Name	Type	TTL	Value
@	A	3600	-
@	NS	172800	ns1-08.azure-dns.com. ns2-08.azure-dns.net. ns3-08.azure-dns.org. ns4-08.azure-dns.info.
@	SOA	3600	Email: azuredns-hostmaster.microsoft... Host: ns1-08.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1
@	MX	3600	0 mail.sysadminsolutions.cf
s1_domainkey	CNAME	3600	s1.domainkey.u26764095.wl096.sendg...
s2_domainkey	CNAME	3600	s2.domainkey.u26764095.wl096.sendg...
auth	CNAME	3600	sysadminsolutions.cf
blog	CNAME	3600	sysadminsolutions.cf
em4764	CNAME	3600	u26764095.wl096.sendgrid.net
ftp	CNAME	3600	sysadminsolutions.cf
ldap	CNAME	3600	sysadminsolutions.cf
mail	CNAME	3600	sysadminsolutions.cf
s1_domainkey.mail	CNAME	3600	s1.domainkey.u26764095.wl096.sendg...
s2_domainkey.mail	CNAME	3600	s2.domainkey.u26764095.wl096.sendg...
em3116.mail	CNAME	3600	u26764095.wl096.sendgrid.net
manager	CNAME	3600	sysadminsolutions.cf
www	CNAME	3600	sysadminsolutions.cf

También abrimos en nuestro firewall los puertos necesarios para que todos los servicios puedan funcionar correctamente:

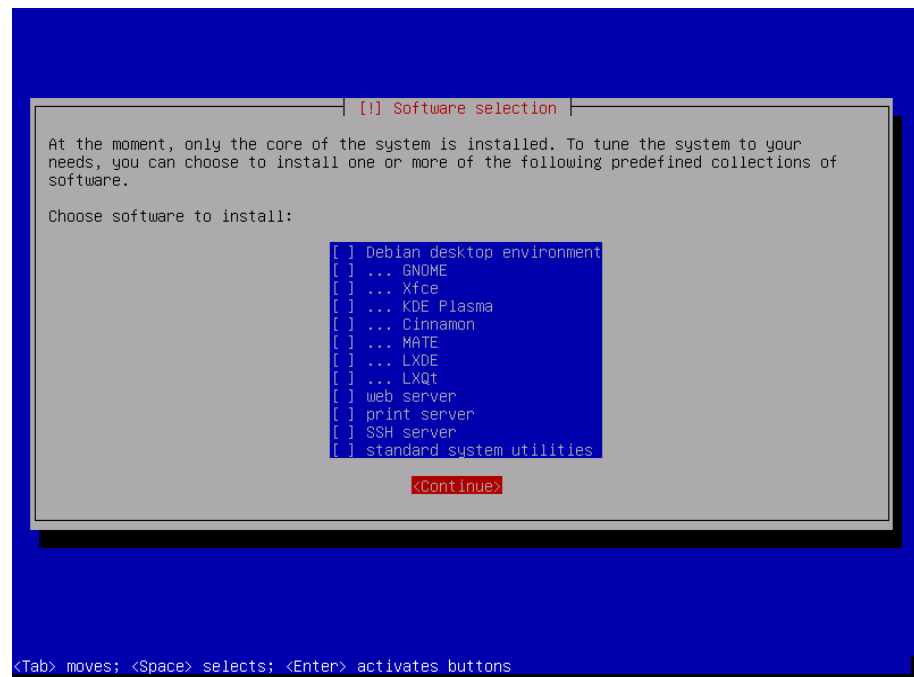
Debian VM SSH	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 22	Forward to lan IP 192.168.1.160 port 22	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM HTTP	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 80	Forward to lan IP 192.168.1.160 port 80	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM HTTPS	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 443	Forward to lan IP 192.168.1.160 port 443	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM SMTP	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 587	Forward to lan IP 192.168.1.160 port 587	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM SMTPS	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 465	Forward to lan IP 192.168.1.160 port 465	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM IMAP	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 143	Forward to lan IP 192.168.1.160 port 143	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM IMAPS	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 993	Forward to lan IP 192.168.1.160 port 993	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM POP	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 110	Forward to lan IP 192.168.1.160 port 110	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM POPS	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 995	Forward to lan IP 192.168.1.160 port 995	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>
Debian VM LDAP	Incoming IPv4, protocol <i>TCP</i> From wan To this device, port 389	Forward to lan IP 192.168.1.160 port 389	<input checked="" type="checkbox"/>	 <a href="#">Edit</a> <a href="#">Delete</a>

Ahora se procederá a la instalación del sistema operativo del servidor, que es una versión mínima de Debian 10.

A la hora de instalarlo seleccionamos la opción Install y posteriormente nuestra localización en España.



Cuando nos pregunte por el hostname introduciremos sysadminsolutions.cf, cuando nos pregunte por una contraseña de root la dejaremos en blanco para no permitir el login directo por root, rellenaremos el nombre de usuario que nos pida a continuación y también la contraseña para este usuario. Cuando nos salga la pantalla para instalar software adicional desmarcamos todas las casillas.



Por último, seleccionamos la partición principal para instalar ahí GRUB y reiniciamos el sistema.

En nuestro primer login instalaremos el servidor de SSH:

```
sudo apt install openssh-server
```

A partir de este momento realizaremos todas las operaciones a través de SSH.

El runlevel por defecto es 5 y lo dejaremos así ya que necesitamos funcionalidades para ofrecer acceso a múltiples usuarios. Al ser una instalación completamente limpia los servicios en ejecución son los indispensables para que el sistema operativo funcione correctamente por lo que no necesitaremos deshabilitar ninguno, estos servicios se pueden comprobar con “systemctl --state=running”.

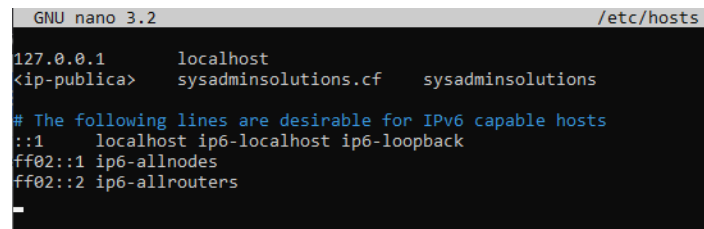
UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
init.scope	loaded	active	running	System and Service Manager
session-2.scope	loaded	active	running	Session 2 of user alberto
cron.service	loaded	active	running	Regular background program processing daemon
dbus.service	loaded	active	running	D-Bus System Message Bus
getty@tty1.service	loaded	active	running	Getty on tty1
getty@tty2.service	loaded	active	running	Getty on tty2
getty@tty3.service	loaded	active	running	Getty on tty3
getty@tty4.service	loaded	active	running	Getty on tty4
getty@tty5.service	loaded	active	running	Getty on tty5
getty@tty6.service	loaded	active	running	Getty on tty6
rsyslog.service	loaded	active	running	System Logging Service
ssh.service	loaded	active	running	OpenBSD Secure Shell server
systemd-journald.service	loaded	active	running	Journal Service
systemd-logind.service	loaded	active	running	Login Service
systemd-timesyncd.service	loaded	active	running	Network Time Synchronization
systemd-udevd.service	loaded	active	running	udev Kernel Device Manager
user@1000.service	loaded	active	running	User Manager for UID 1000
dbus.socket	loaded	active	running	D-Bus System Message Bus Socket
syslog.socket	loaded	active	running	Syslog Socket
systemd-journald-audit.socket	loaded	active	running	Journal Audit Socket
systemd-journald-dev-log.socket	loaded	active	running	Journal Socket (/dev/log)
systemd-journald.socket	loaded	active	running	Journal Socket
systemd-udevd-control.socket	loaded	active	running	udev Control Socket
systemd-udevd-kernel.socket	loaded	active	running	udev Kernel Socket

## 2. Descripción del sistema y sus servicios

### 2.1. Hostname y MOTD

Ya hemos configurado el hostname del servidor durante la instalación de Debian, pero modificaremos el archivo `/etc/hosts` para que el nombre de nuestro dominio apunte internamente a nuestra IP pública:

```
sudo nano /etc/hosts
```

A screenshot of the GNU nano 3.2 text editor showing the contents of the /etc/hosts file. The file contains mappings for 127.0.0.1 to localhost, a placeholder <ip-publica> to sysadminsolutions.cf and sysadminsolutions, and IPv6 addresses to localhost, ip6-loopback, ip6-allnodes, and ip6-allrouters.

```
GNU nano 3.2 /etc/hosts
127.0.0.1    localhost
<ip-publica> sysadminsolutions.cf  sysadminsolutions

# The following lines are desirable for IPv6 capable hosts
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Para actualizar el MOTD podemos editar el archivo `/etc/motd` y poner lo que queramos ahí:

A screenshot of the GNU nano 3.2 text editor showing the contents of the /etc/motd file. It features a large ASCII art graphic of a smiley face. At the bottom, the text 'No hagas nada estúpido' is displayed.

```
GNU nano 3.2 /etc/motd

      ,met$$$$$gg.
    ,g$$$$$$$$$$$$P.
  ,g$$$P""""""Y$$$.
 ,$$P'         `$$$$.
 '$,$$P       ,gg$.  `$$b:
 `d$$'      ,d$P""  . $$$
  $$P      d$'     , $$P
 $$:      $$.     - ,d$$'
 $$;      Y$b._   ,d$P'
 Y$$.     `."Y$$$$$P"
 `$$b     "-._
  `Y$$b
   `Y$$.
    `$$b.
     `Y$$b.
      ``Y$b._
        ~~~~~

No hagas nada estúpido
```

### 2.2. Servidor web

Para el servidor web instalamos el paquete de Nginx:

```
sudo apt install -y nginx nginx-extras
```

En el archivo `/etc/nginx/nginx.conf` descomentaremos la línea “`# server_names_hash_bucket_size 64;`” para evitar futuros problemas:

```
sudo sed -i 's/# server_names_hash_bucket_size
64;/server_names_hash_bucket_size 64;/g' /etc/nginx/nginx.conf
```

Además, necesitaremos también un servidor de bases de datos, para ello instalamos MariaDB, usaremos la versión 10.6 ya que es la versión LTS más reciente:

```
sudo apt install wget curl
wget https://downloads.mariadb.com/MariaDB/mariadb_repo_setup
chmod +x mariadb_repo_setup
sudo ./mariadb_repo_setup --mariadb-server-version="mariadb-10.6"
sudo apt update
sudo apt install -y mariadb-server
rm mariadb_repo_setup
```

Para configurar algunos aspectos de la seguridad de la base de datos ejecutamos la orden:

```
sudo mysql_secure_installation
```

Aquí no configuramos ninguna contraseña para el root de MariaDB, y para el resto de las opciones seleccionaremos los valores de defecto.

Para poder procesar código desde una página web en el servidor necesitamos añadir soporte para PHP. Instalaremos los paquetes correspondientes:

```
sudo apt install -y php-fpm php-mysql
```

### 2.3. Usuarios

Para hacer que los usuarios dispongan del mismo login para usar todos los servicios instalaremos un servidor de LDAP en nuestra máquina, en este caso usaremos OpenLDAP:

```
sudo apt install -y slapd ldap-utils ldapscripts
```

Por ahora podemos dejar la contraseña vacía ya que la configuraremos ahora de nuevo junto con otros aspectos, con la orden:

```
sudo dpkg-reconfigure slapd
```

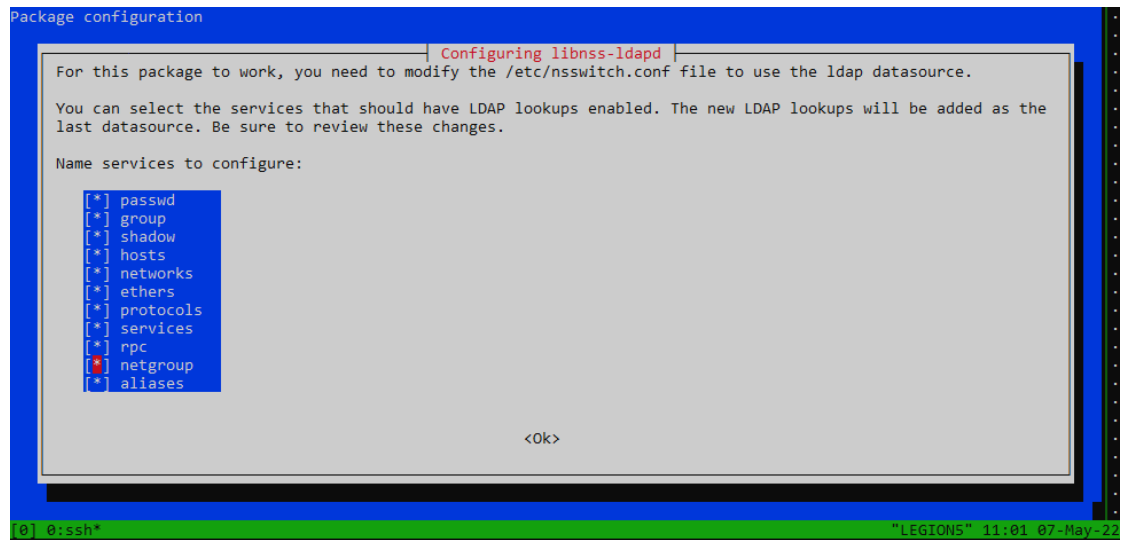
Aquí crearemos la configuración inicial del servidor LDAP, cuando nos pidan el nombre del dominio DNS introduciremos "sysadminsolutions.cf", el nombre de la organización también será "sysadminsolutions.cf", ahora pondremos la contraseña de administración de LDAP. Para el resto de las opciones seleccionaremos los valores de defecto.

Para permitir que los usuarios añadidos a LDAP puedan hacer login en nuestro sistema Linux instalaremos los paquetes que añaden soporte de LDAP para NSS y PAM:

```
sudo apt install -y libnss-ldapd libpam-ldapd
```

Cuando el configurador nos pregunte por la URI del servidor LDAP introduciremos la dirección "ldap://sysadminsolutions.cf". Cuando nos pregunte por la base de búsqueda del servidor LDAP introduciremos "dc=sysadminsolutions,dc=cf". Cuando nos pregunte qué servicios queremos que usen la base de datos de LDAP seleccionaremos todos los disponibles.

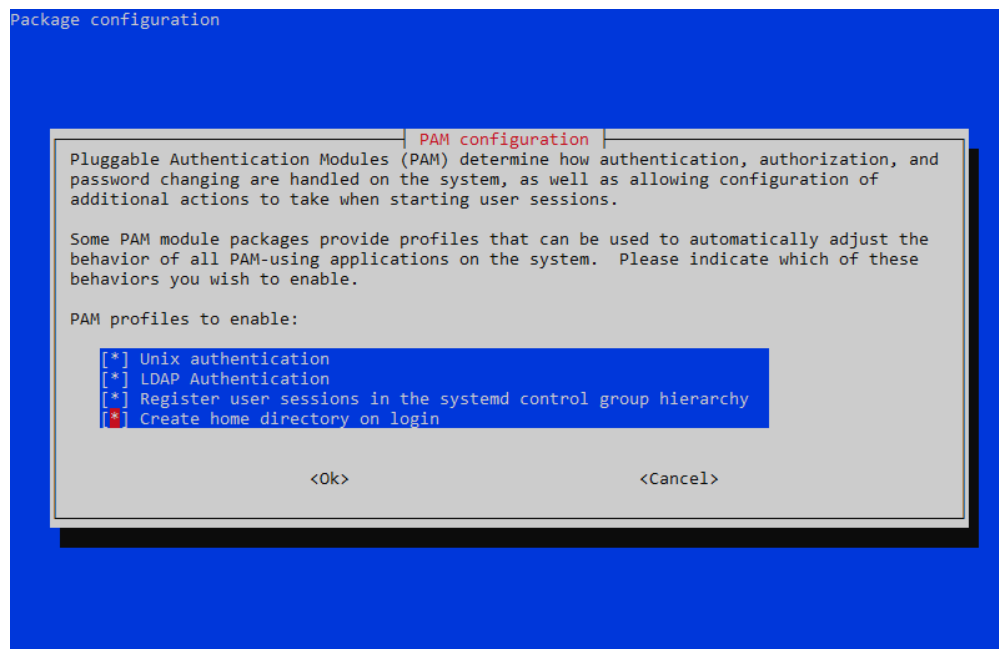




Para que se cree el directorio home de los usuarios cuando inicien sesión por primera vez ejecutaremos el comando:

```
sudo pam-auth-update
```

Y marcaremos todas las opciones disponibles:



Además, para que el directorio home del usuario se cree con permisos 700 (los usuarios que no sean el propietario no tendrán ningún permiso ni de lectura ni de ejecución) editaremos el archivo `/etc/pam.d/common-session` e indicaremos la máscara adecuada después de `pam_mkhomedir.so`.

```
session optional                                pam_mkhomedir.so umask=0077
```

Para administrar los usuarios de LDAP desde una interfaz web instalaremos FusionDirectory:

```
sudo apt install -y fusiondirectory fusiondirectory-schema
```

Y ejecutaremos la siguiente orden para añadir los esquemas necesarios a la base de datos de LDAP:

```
sudo fusiondirectory-insert-schema
```

Para poder añadir usuarios Unix desde la interfaz web y permitir a los usuarios que editen su correo electrónico instalaremos los siguientes paquetes:

```
sudo apt install fusiondirectory-plugin-posix fusiondirectory-plugin-mail fusiondirectory-plugin-mail-schema
sudo fusiondirectory-insert-schema -i /etc/ldap/schema/fusiondirectory/mail-fd.schema
sudo fusiondirectory-insert-schema -i /etc/ldap/schema/fusiondirectory/mail-fd-conf.schema
```

Ahora añadiremos un bloque de Nginx para acceder a la página de administración de LDAP desde `ldap.sysadminsolutions.cf`:

```
sudo tee /etc/nginx/sites-available/ldap.sysadminsolutions.cf > /dev/null <<'EOF'
server {
    listen 80;
    root /usr/share/fusiondirectory/html;
    index index.php index.html index.htm;
    server_name ldap.sysadminsolutions.cf;

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
    }

    location / {
        try_files $uri $uri/ =404;
    }
}
EOF
```

Ahora activaremos el sitio haciendo un enlace simbólico:

```
sudo ln -s /etc/nginx/sites-available/ldap.sysadminsolutions.cf /etc/nginx/sites-enabled/
```

Probaremos que la configuración es correcta:

```
sudo nginx -t
```

Y reiniciamos el servicio de Nginx para hacer visibles los cambios:

```
sudo systemctl restart nginx
```

Para asegurar la conexión a LDAP obtendremos un certificado SSL usando la herramienta Certbot, para ello instalaremos los paquetes necesarios:

```
sudo apt install -y python3-acme python3-certbot python3-mock python3-openssl python3-pkg-resources python3-pyparsing python3-zope.interface python3-certbot-nginx
```

Y obtendremos un certificado SSL para la dirección `ldap.sysadminsolutions.cf` con la siguiente orden:

```
sudo certbot --redirect --nginx -d ldap.sysadminsolutions.cf
```

Los certificados generados se guardan en `/etc/letsencrypt/live/ldap.sysadminsolutions.cf/`

Para poder usar los certificados con LDAP crearemos un directorio donde copiaremos los certificados y les asignaremos permisos adecuados:

```
sudo mkdir -p /etc/ldap/ssl
sudo cp /etc/letsencrypt/live/ldap.sysadminsolutions.cf/fullchain.pem /etc/ldap/ssl/fullchain.crt
sudo cp /etc/letsencrypt/live/ldap.sysadminsolutions.cf/cert.pem /etc/ldap/ssl/ldap.crt
sudo cp /etc/letsencrypt/live/ldap.sysadminsolutions.cf/privkey.pem /etc/ldap/ssl/ldap.key
sudo chown -R openldap:openldap /etc/ldap/ssl
```

Para evitar tener que ejecutar estos comandos después de cada renovación del certificado haremos que certbot lo haga de forma automática creando un script en `/etc/letsencrypt/renewal-hooks/post/`:

```
sudo tee /etc/letsencrypt/renewal-hooks/post/update-owner.sh > /dev/null <<EOF
#!/bin/bash
sudo cp /etc/letsencrypt/live/ldap.sysadminsolutions.cf/fullchain.pem /etc/ldap/ssl/fullchain.crt
sudo cp /etc/letsencrypt/live/ldap.sysadminsolutions.cf/cert.pem /etc/ldap/ssl/ldap.crt
sudo cp /etc/letsencrypt/live/ldap.sysadminsolutions.cf/privkey.pem /etc/ldap/ssl/ldap.key
sudo chown -R openldap:openldap /etc/ldap/ssl
EOF

sudo chmod o+x /etc/letsencrypt/renewal-hooks/post/update-owner.sh
```

Añadiremos el certificado a la configuración de LDAP con la siguiente orden:

```
sudo ldapmodify -H ldapi:/// -Y EXTERNAL << EOF
dn: cn=config
changetype: modify
replace: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/ssl/fullchain.crt
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/ssl/ldap.crt
```

```
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/ssl/ldap.key
-
replace: olcTLSVerifyClient
olcTLSVerifyClient: never
EOF
```

Además, podemos forzar al servidor LDAP a usar conexiones a través de TLS:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<'EOF'
dn: cn=config
changetype: modify
add: olcSecurity
olcSecurity: tls=1
EOF
```

Añadimos la siguiente línea a `/etc/ldap/ldap.conf` para omitir la verificación del certificado:

```
sudo tee -a /etc/ldap/ldap.conf > /dev/null <<'EOF'
TLS_REQCERT      never
EOF
```

Reiniciamos el servicio de LDAP:

```
sudo systemctl restart slapd
```

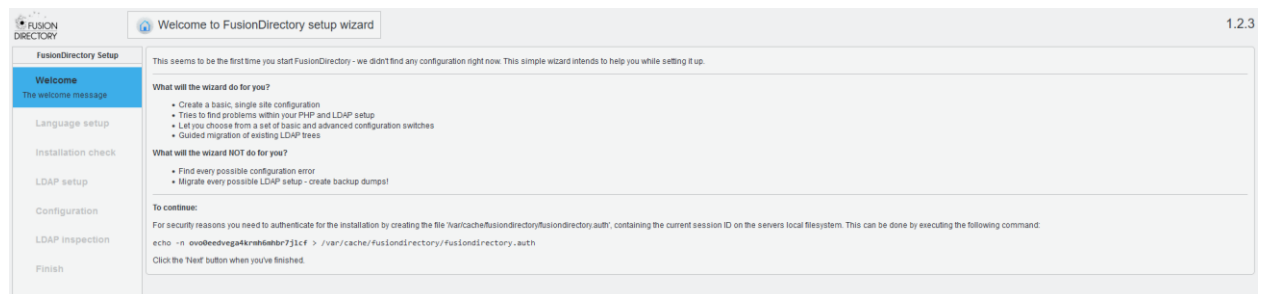
Ahora vamos a reconfigurar el servicio `nsld` para que pueda conectarse al servidor LDAP de forma encriptada:

```
sudo dpkg-reconfigure nsld
```

Cuando nos pregunte el método de autenticación seleccionaremos simple, luego introducimos “`cn=admin,dc=sysadminsolutions,dc=cf`” y la contraseña y después cuando nos pregunte si deseamos usar StartTLS, seleccionaremos Sí. Por último, reiniciamos el servicio `nsd` y `nsld`:

```
sudo systemctl restart nsd nsld
```

Ahora seguiremos la configuración accediendo a la dirección <https://ldap.sysadminsolutions.cf/>



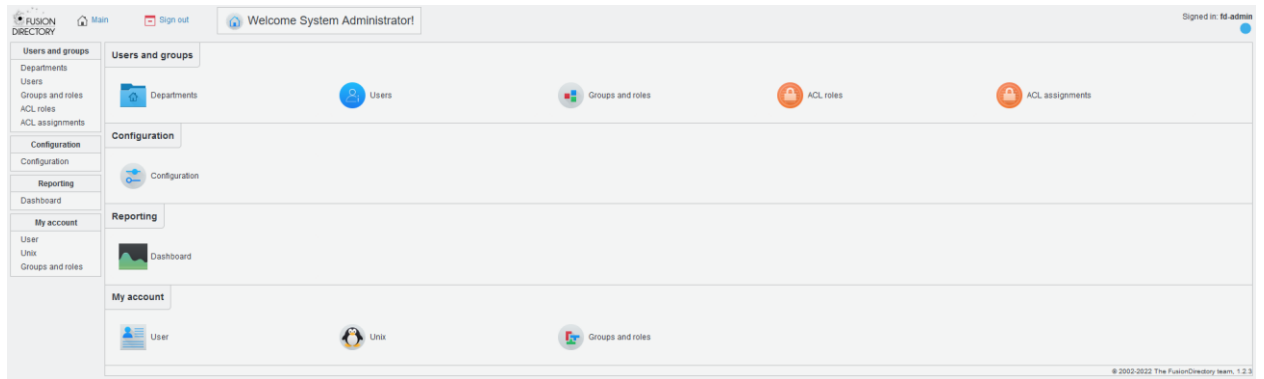
En el paso de LDAP setup tendremos que introducir la contraseña del administrador de LDAP, introducir la URI de conexión “`ldap://ldap.sysadminsolutions.cf:389`”, marcar la casilla de conexión TLS e introducir en el campo Base “`dc=sysadminsolutions,dc=cf`”.

En el siguiente paso indicaremos el directorio del certificado SSL y su clave (/etc/letsencrypt/live/ldap.sysadminsolutions.cf/fullchain.pem y /etc/letsencrypt/live/ldap.sysadminsolutions.cf/privkey.pem).

En el último paso aplicaremos las configuraciones recomendadas, también crearemos un usuario en LDAP para nuestro administrador. Una vez finalizada la configuración hay que descargar el archivo de configuración fusiondirectory.conf a través de SFTP y colocarlo en el directorio /etc/fusiondirectory/. Para asignar los permisos adecuados ejecutaremos la orden:

```
sudo fusiondirectory-setup --check-config
```

Ahora podemos hacer login con las credenciales introducidas anteriormente.



Desde aquí ahora podemos añadir grupos y usuarios para comprobar que todo funciona correctamente, desde la pestaña de Groups and roles iremos a Actions y seleccionaremos Create > POSIX Group y crearemos los grupos de alumnos y profesores, desde la pestaña de Users ahora podemos crear un usuario de prueba, vamos a Actions > Create > User y rellenamos los campos con el nombre real del usuario, su login, su contraseña, y en la pestaña de Unix le asignamos su directorio home, la shell, y el grupo al que pertenece.

Al hacer `getent passwd` y `getent group` podemos comprobar que el usuario y grupo se han creado correctamente:

```
alberto@sysadminsolutions:~$ getent passwd | grep pgarcia
pgarcia:x:1102:1101:Garcia Pepe:/home/pgarcia:/bin/bash
alberto@sysadminsolutions:~$ getent group | grep alumnos
alumnos:*:1101:pgarcia
```

Para que todos los usuarios puedan editar su información de usuario y contraseña, desde FusionDirectory iremos a ACL assignments > sysadminsolutions y añadiremos un nuevo Assignment indicando el rol “editowninfos” y marcando la casilla de “For all users” y guardamos el cambio. También editaremos el rol para que no puedan modificar los datos de su cuenta UNIX.

## 2.4. Portal web

Para que nuestros usuarios puedan acceder a nuestros servicios desde una página web, instalaremos el paquete `LemonLDAP::NG` que nos proporcionará la posibilidad de que los usuarios hagan login directamente a los servicios con su usuario de LDAP:

```
sudo apt install -y apt-transport-https gnupg
wget -O - https://lemonldap-ng.org/_media/rpm-gpg-key-ow2 | sudo apt-
key add -
```

```
sudo tee /etc/apt/sources.list.d/lemonldap-ng.list > /dev/null <<'EOF'
deb https://lemonldap-ng.org/deb stable main
EOF
sudo apt update
sudo apt install -y lemonldap-ng
```

Para configurar LemonLDAP y que apunte a nuestro servidor hay que sustituir el dominio que viene por defecto, `example.com`, por nuestro dominio, `sysadminsolutions.cf`:

```
sudo sed -i 's/example\.com/sysadminsolutions.cf/g' /etc/lemonldap-ng/*
/var/lib/lemonldap-ng/conf/lmConf-1.json /etc/nginx/sites-available/*
```

Además, cambiaremos los archivos necesarios para acceder al portal desde el dominio principal en vez desde `auth.sysadminsolutions.cf`:

```
sudo sed -i 's/auth\.sysadminsolutions\.cf/sysadminsolutions.cf/g'
/etc/lemonldap-ng/* /var/lib/lemonldap-ng/conf/lmConf-1.json
/etc/nginx/sites-available/*
```

Para poder usar LemonLDAP con Nginx hay que activar el servicio lemonldap-ng-fastcgi-server y las páginas web de LemonLDAP:

```
cd /etc/nginx/sites-enabled
sudo ln -s /etc/lemonldap-ng/nginx-lua-headers.conf
/etc/nginx/snippets/llng-lua-headers.conf
sudo ln -s ../sites-available/*nginx* .
cd ~
```

Para evitar problemas podemos añadir la línea “variables\_hash\_max\_size 2048;” al inicio del archivo /etc/nginx/sites-available/manager-nginx.conf

Además, eliminaremos el dominio que viene por defecto con Nginx y reiniciamos el servicio:

```
sudo rm /etc/nginx/sites-enabled/default
sudo systemctl restart llng-fastcgi-server nginx
```

Luego crearemos certificados SSL para acceder al portal a través de HTTPS:

```
sudo certbot --redirect --nginx -d sysadminsolutions.cf -d
www.sysadminsolutions.cf -d manager.sysadminsolutions.cf
```

Ahora podemos finalizar la instalación a través de la dirección:

<https://manager.sysadminsolutions.cf>, sin embargo seguiremos realizando la configuración a través de la terminal.

Para indicar la dirección URL del portal y forzar HTTPS usaremos el siguiente comando:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
set \
portal https://sysadminsolutions.cf/ \
mailUrl https://sysadminsolutions.cf/resetpwd \
registerUrl https://sysadminsolutions.cf/register \
https 1 \
securedCookie 1
```

Para configurar los parámetros de LDAP usaremos la siguiente orden:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
set \
authentication LDAP \
userDB LDAP \
passwordDB LDAP \
registerDB LDAP \
ldapServer 'ldap+tls://sysadminsolutions.cf' \
ldapVerify 'none' \
managerDn 'cn=admin,dc=sysadminsolutions,dc=cf' \
managerPassword 'dqAehD7UUrM7C0' \
ldapBase 'ou=people,dc=sysadminsolutions,dc=cf' \
ldapPolicyControl 1
```

Seleccionaremos los parámetros que deseamos exportar de LDAP:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
    ldapExportedVars uid uid \
    ldapExportedVars cn cn \
    ldapExportedVars sn sn \
    ldapExportedVars mobile mobile \
    ldapExportedVars mail mail \
    ldapExportedVars givenName givenName
```

Para permitir sólo al administrador acceder a la página web de administración de LemonLDAP eliminaremos las reglas de acceso por defecto y añadiremos una excepción:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  delKey \
    locationRules/manager.sysadminsolutions.cf
'(?#Configuration)^(/*?\. (fcgi|psgi)/)?(manager\.html|confs|prx/|$)' \
  locationRules/manager.sysadminsolutions.cf
'(?#Sessions)/(/*?\. (fcgi|psgi)/)?sessions' \
  locationRules/manager.sysadminsolutions.cf
'(?#Notifications)/(/*?\. (fcgi|psgi)/)?notifications'

sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
    locationRules/manager.sysadminsolutions.cf default '$uid =~
/^alberto$/'
```

Ahora vinculamos FusionDirectory con LemonLDAP para poder acceder directamente desde el portal sin tener que volver a introducir nuestras credenciales. Para ello primero añadimos el dominio a los virtual hosts:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
    'locationRules/ldap.sysadminsolutions.cf' 'default' 'accept' \
    'locationRules/ldap.sysadminsolutions.cf'
'(?#Logout)~/index\.php?signout=1' 'logout_sso' \
    'exportedHeaders/ldap.sysadminsolutions.cf' 'Auth-User' '$uid'
```

Y modificaremos el archivo `/etc/nginx/sites-available/ldap.sysadminsolutions.cf` reemplazando la sección de location `\.php$` existente por:

```
location = /lmauth {
    internal;
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/llng-fastcgi-server/llng-fastcgi.sock;
    fastcgi_pass_request_body off;
    fastcgi_param CONTENT_LENGTH "";
    fastcgi_param HOST $http_host;
    fastcgi_param X_ORIGINAL_URI $original_uri;
}

location ~ /\.php$ {
    auth_request /lmauth;
```



```

set $original_uri $uri$sis_args$args;
auth_request_set $lmremote_user $upstream_http_lm_remote_user;
auth_request_set $lmlocation $upstream_http_location;
auth_request_set $cookie_value $upstream_http_set_cookie;
add_header Set-Cookie $cookie_value;
error_page 401 $lmlocation;
include snippets/fastcgi-php.conf;
fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
include snippets/lmg-lua-headers.conf;
}

```

Reiniciamos Nginx:

```
sudo nginx -t && sudo systemctl restart nginx
```

En `ldap.sysadminsolutions.cf` vamos a Configuration y marcamos la casilla de HTTP header authentication. Ahora al hacer logout nos llevará a la página de login del portal.

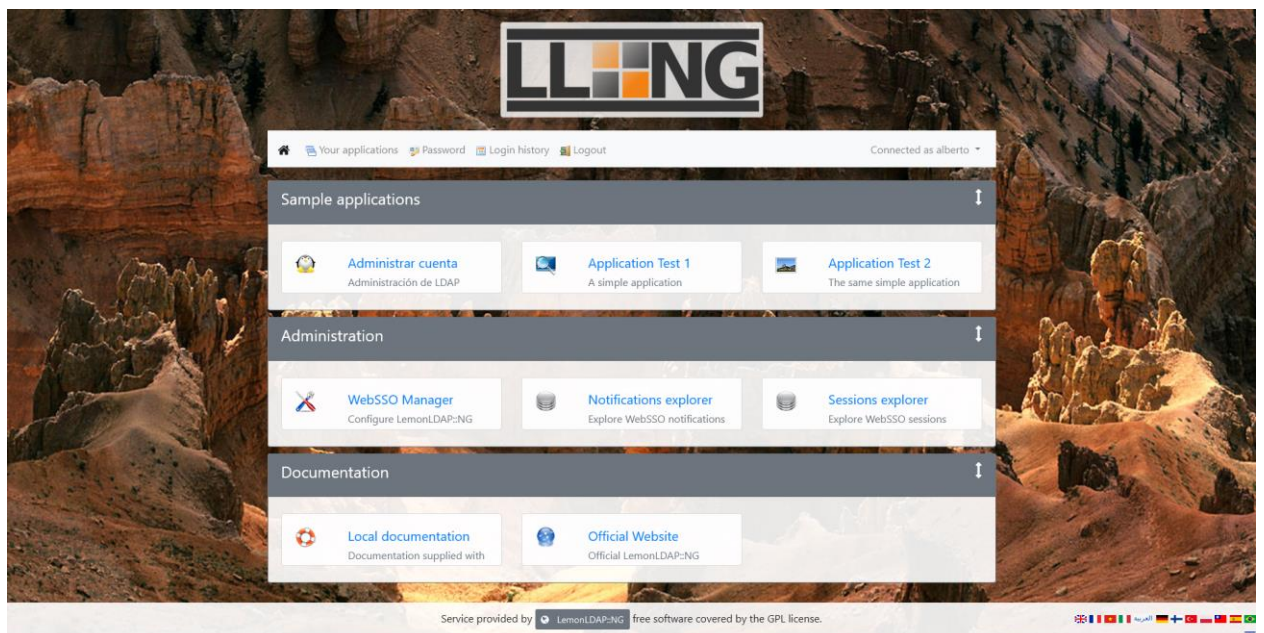
Para añadir FusionDirectory a la lista de apps del portal usaremos la siguiente orden:

```

sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
  applicationList/1sample/ldap type application \
  applicationList/1sample/ldap/options description "Haz cambios en LDAP" \
  applicationList/1sample/ldap/options display "on" \
  applicationList/1sample/ldap/options logo "tux.png" \
  applicationList/1sample/ldap/options name "Administrar cuenta" \
  applicationList/1sample/ldap/options uri
"https://ldap.sysadminsolutions.cf/"

```

Ahora ya podemos acceder a FusionDirectory desde el portal sin tener que volver a hacer login:



También configuraremos LemonLDAP como un proveedor de identidad CAS para más tarde poder usarlo para proporcionar Single-Sign-On a más servicios:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \  
set \  
    issuerDBCASActivation 1 \  
    casTicketExpiration 300 \  
    casAccessControlPolicy error
```

Activamos el botón de restablecer contraseña en la pantalla de inicio de sesión:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \  
set \  
    portalDisplayResetPassword 1
```

Configuraremos algunos parámetros para ajustar la apariencia de LemonLDAP, como cambiar el logo, el fondo y eliminar las aplicaciones que aparecen por defecto.

```
sudo wget https://i.imgur.com/XLRin63.png -O /usr/share/lemonldap-  
ng/portal/htdocs/static/common/logos/USAL-Logo.png  
sudo wget https://i.imgur.com/IcziPOi.jpg -O /usr/share/lemonldap-  
ng/portal/htdocs/static/common/backgrounds/background.jpg  
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \  
set \  
    portalMainLogo 'common/logos/USAL-Logo.png' \  
    portalSkinBackground 'background.jpg'
```

Por defecto LemonLDAP proporciona una forma simple para consultar quién ha accedido o ha intentado acceder a la cuenta del usuario yendo a la pestaña de Historial de conexión:



UNIVERSIDAD DE SALAMANCA

Sus aplicaciones Contraseña Historial de conexión Desconexión Conectado como alberto

### Últimas conexiones

Fecha	Dirección IP
5/22/2022, 2:55:35 AM	192.168.1.1
5/21/2022, 11:43:00 PM	192.168.1.1
5/21/2022, 7:33:48 PM	192.168.1.1
5/21/2022, 3:24:40 PM	192.168.1.1
5/21/2022, 1:08:14 PM	192.168.1.1

Last logins

### Últimas conexiones fallidas

Fecha	Dirección IP	Mensaje de Error
5/21/2022, 3:24:37 PM	192.168.1.1	Contraseña o identificador incorrecto
5/21/2022, 3:24:33 PM	192.168.1.1	Contraseña o identificador incorrecto

### 2.4.1. Registro de usuarios

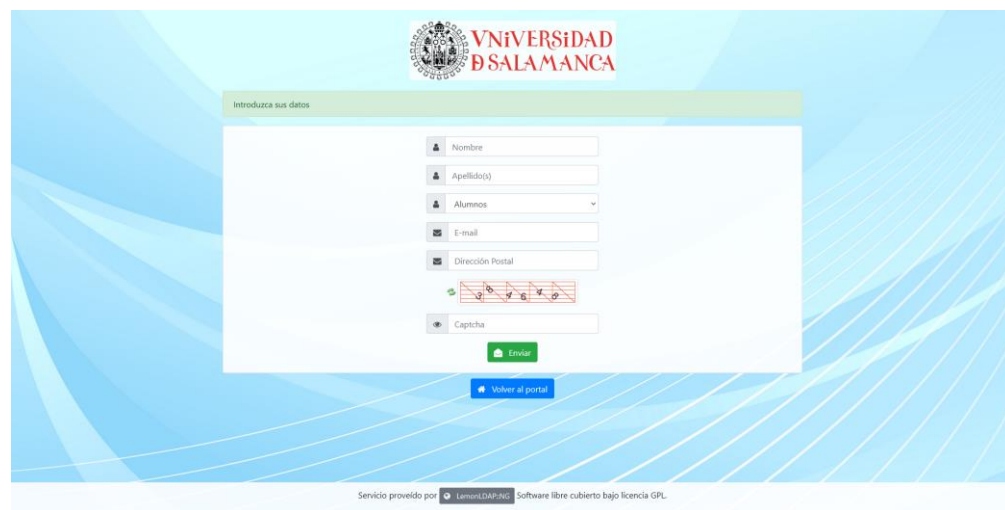
LemonLDAP incluye por defecto la posibilidad de crear usuarios en LDAP a partir de un formulario web, enviando al usuario un correo electrónico de confirmación y posteriormente otro incluyendo sus credenciales, pero hay que hacer algunos cambios en su código para cumplir con los requisitos de la práctica.

Primero modificamos el script de Perl que se encarga de pasar los datos a LDAP para poder indicar que las cuentas creadas también son cuentas de usuarios Unix junto con los parámetros necesarios. Posteriormente, el script de Perl que se encarga de pasar los datos del HTML al script de LDAP y por último el HTML en sí, adjuntamos el archivo LemonLDAP.patch con los cambios incluidos, se puede aplicar con:

```
sudo apt install -y patch
sudo patch -d/ -p0 < LemonLDAP.patch
sudo apt install cpanminus build-essential
sudo cpanm Digest::SHA1 MIME::Base64
rm LemonLDAP.patch
```

Una vez aplicados los cambios reiniciamos LemonLDAP:

```
sudo systemctl restart llng-fastcgi-server
```



## 2.5. Correo electrónico

Para el servidor de correo electrónico instalamos Postfix:

```
sudo apt install -y postfix
```

En las opciones de configuración seleccionamos las opciones por defecto.

Ahora instalaremos Dovecot como servidor IMAP y POP3:

```
sudo apt install -y dovecot-core dovecot-imapd dovecot-pop3d
```

Una vez instalado configuraremos Dovecot para que use el sistema de almacenamiento de emails Maildir y los guarde en ~/.maildir:

```
sudo sed -i 's/mail_location =
mailbox:~/mail:INBOX=~/.maildir/g' /etc/dovecot/conf.d/10-mail.conf
sudo adduser dovecot mail
```

Ahora haremos que Postfix relegue en Dovecot el almacenamiento de mensajes:

```
sudo apt install dovecot-lmtpd
sudo nano /etc/dovecot/conf.d/10-master.conf
```

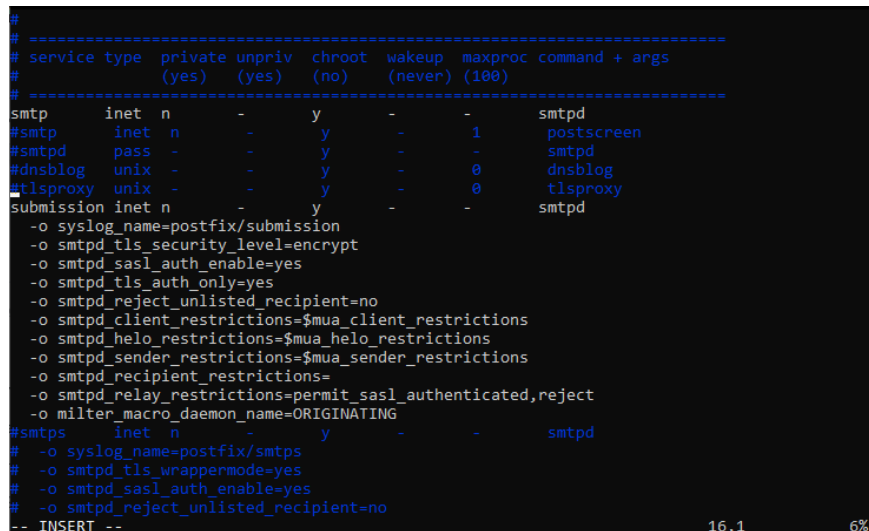
Sustituiremos la sección service lmtp por lo siguiente:

```
service lmtp {
    unix_listener /var/spool/postfix/private/dovecot-lmtp {
        mode = 0600
        user = postfix
        group = postfix
    }
}
```

Y añadimos la configuración en Postfix

```
sudo postconf -e 'mailbox_transport = lmtp:unix:private/dovecot-lmtp'
sudo postconf -e 'smtpUTF8_enable = no'
```

Posteriormente habilitaremos el servicio submission de Postfix, para ello editaremos el fichero /etc/postfix/master.cf y descomentaremos las líneas de la sección submission.



```
#
# =====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (no)   (never) (100)
# =====
smtp      inet  n       -       y       -       -       smtpd
#smtp     inet  n       -       y       -       1       postscreen
#smtpd    pass  -       -       y       -       -       smtpd
#dnsblog  unix  -       -       y       -       0       dnsblog
#tlsproxy unix  -       -       y       -       0       tlsproxy
submission inet n       -       y       -       -       smtpd
-o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_tls_auth_only=yes
-o smtpd_reject_unlisted_recipient=no
-o smtpd_client_restrictions=$mua_client_restrictions
-o smtpd_helo_restrictions=$mua_helo_restrictions
-o smtpd_sender_restrictions=$mua_sender_restrictions
-o smtpd_recipient_restrictions=
-o smtpd_relay_restrictions=permit_sasl_authenticated,reject
-o milter_macro_daemon_name=ORIGINATING
#smtps     inet  n       -       y       -       -       smtpd
#-o syslog_name=postfix/smtps
#-o smtpd_tls_wrappermode=yes
#-o smtpd_sasl_auth_enable=yes
#-o smtpd_reject_unlisted_recipient=no
-- INSERT --
```

Para que Postfix pueda acceder al servicio de autenticación de Dovecot, editaremos el archivo /etc/dovecot/conf.d/10-master.conf y sustituimos la sección de “# Postfix smtp-auth” por:

```
unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
}
```

```
    group = postfix
}
```

En el lado de Postfix ejecutaremos las siguientes órdenes para configurar la autenticación a través de Dovecot:

```
sudo postconf -e 'smtpd_sasl_type = dovecot'
sudo postconf -e 'smtpd_sasl_path = private/auth'
sudo postconf -e 'smtpd_sasl_local_domain ='
sudo postconf -e 'smtpd_sasl_security_options = noanonymous'
sudo postconf -e 'smtpd_sasl_tls_security_options = noanonymous'
sudo postconf -e 'broken_sasl_auth_clients = yes'
sudo postconf -e 'smtpd_sasl_auth_enable = yes'
sudo postconf -e 'smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
```

También editaremos el archivo `/etc/dovecot/conf.d/10-auth.conf` para eliminar el dominio del nombre de usuario si se recibe al hacer login y para añadir compatibilidad con Outlook.

```
sudo sed -i 's/#auth_username_format = %Lu/auth_username_format = %n/g'
/etc/dovecot/conf.d/10-auth.conf
sudo sed -i 's/auth_mechanisms = plain/auth_mechanisms = plain login/g'
/etc/dovecot/conf.d/10-auth.conf
```

Para nuestro cliente de correo instalaremos la última versión de Roundcube:

```
wget
https://github.com/roundcube/roundcubemail/releases/download/1.5.2/roundcubemail-1.5.2-complete.tar.gz
tar xvf roundcubemail-1.5.2-complete.tar.gz
rm roundcubemail-1.5.2-complete.tar.gz
sudo mv roundcubemail-1.5.2 /var/www/roundcube
```

Instalaremos las dependencias de Roundcube:

```
sudo apt install -y php7.3-ldap php-imagick php7.3-common php7.3-gd
php7.3-imap php7.3-json php7.3-curl php7.3-zip php7.3-xml php7.3-
mbstring php7.3-bz2 php7.3-intl php7.3-gmp
```

Cambiamos los permisos de los directorios a los que el servidor web necesita acceder:

```
sudo chown www-data:www-data /var/www/roundcube/temp/
/var/www/roundcube/logs/ -R
```

Ahora crearemos una base de datos y usuario en MariaDB para Roundcube:

```
sudo mysql -u root
CREATE DATABASE roundcube DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
CREATE USER roundcubeuser@localhost IDENTIFIED BY 'xsJytI17hrcCJk';
GRANT ALL PRIVILEGES ON roundcube.* TO roundcubeuser@localhost;
flush privileges;
exit
sudo mysql roundcube < /var/www/roundcube/SQL/mysql.initial.sql
```

A continuación, crearemos un bloque de Nginx para poder acceder a la página de Roundcube:

```
sudo tee /etc/nginx/sites-available/mail.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen 80;
    server_name mail.sysadminsolutions.cf;

    root          /var/www/roundcube/;
    index          index.php;
    charset        utf-8;

    location / {
        try_files $uri $uri/ index.php;
    }

    location ~ ^/(README|INSTALL|LICENSE|CHANGELOG|UPGRADING)$ {
        deny all;
    }

    location ~ ^/(bin|SQL|config|temp|logs)/ {
        deny all;
    }

    location ~* \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.(php))(/.+)$;
        fastcgi_pass    unix:/run/php/php7.3-fpm.sock;
        fastcgi_index    index.php;
        fastcgi_param    SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
EOF

sudo ln -s /etc/nginx/sites-available/mail.sysadminsolutions.cf
/etc/nginx/sites-enabled/
```

**Reiniciamos Nginx:**

```
sudo nginx -t && sudo systemctl reload nginx
```

**Ahora crearemos un certificado para poder acceder a la página a través de HTTPS:**

```
sudo certbot --redirect --nginx -d mail.sysadminsolutions.cf
```

Con estos certificados también podemos hacer que Dovecot emplee una conexión segura a la hora de identificar al usuario. Primero editaremos el archivo `/etc/dovecot/conf.d/10-auth.conf`

y para deshabilitar la autenticación en texto plano, editaremos el fichero `/etc/dovecot/conf.d/10-ssl.conf` para incluir el certificado generado y su clave:

```
sudo sed -i 's/#disable_plaintext_auth = yes/disable_plaintext_auth = yes/g' /etc/dovecot/conf.d/10-auth.conf
sudo sed -i 's/\s\/etc\/dovecot\/private\/dovecot.pem\/etc\/letsencrypt\/live\/mail.sysadminsolutions.cf\/fullchain.pem/g' /etc/dovecot/conf.d/10-ssl.conf
sudo sed -i 's/\s\/etc\/dovecot\/private\/dovecot.key\/etc\/letsencrypt\/live\/mail.sysadminsolutions.cf\/privkey.pem/g' /etc/dovecot/conf.d/10-ssl.conf
```

También cambiaremos unos ajustes para usar protocolos más seguros que los de por defecto:

```
sudo sed -i 's/ssl = yes/ssl = required/g' /etc/dovecot/conf.d/10-ssl.conf
sudo sed -i 's/#ssl_prefer_server_ciphers = no/ssl_prefer_server_ciphers = yes/g' /etc/dovecot/conf.d/10-ssl.conf
sudo sed -i 's/#ssl_min_protocol = TLSv1/ssl_min_protocol = TLSv1.2/g' /etc/dovecot/conf.d/10-ssl.conf
```

También configuraremos los certificados para Postfix con las siguientes órdenes:

```
sudo postconf -e 'smtp_tls_security_level = may'
sudo postconf -e 'smtpd_tls_security_level = may'
sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
sudo postconf -e 'smtpd_tls_key_file = /etc/letsencrypt/live/mail.sysadminsolutions.cf/privkey.pem'
sudo postconf -e 'smtpd_tls_cert_file = /etc/letsencrypt/live/mail.sysadminsolutions.cf/fullchain.pem'
sudo postconf -e 'smtpd_tls_loglevel = 1'
sudo postconf -e 'smtpd_tls_received_header = yes'
sudo postconf -e 'myhostname = mail.sysadminsolutions.cf'
```

Para poder mandar correos electrónicos cuando el puerto 25 está bloqueado, usaremos un servicio de SMTP relay gratuito como SendGrid y configuraremos Postfix adecuadamente:

```
sudo postconf -e 'smtp_sasl_auth_enable = yes'
sudo postconf -e 'smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd'
sudo postconf -e 'smtp_sasl_security_options = noanonymous'
sudo postconf -e 'smtp_sasl_tls_security_options = noanonymous'
sudo postconf -e 'smtp_tls_security_level = encrypt'
sudo postconf -e 'header_size_limit = 4096000'
sudo postconf -e 'relayhost = [smtp.sendgrid.net]:587'
```

En el archivo `/etc/postfix/sasl_passwd` guardaremos la clave API de nuestra cuenta SendGrid:

```
sudo tee /etc/postfix/sasl_passwd > /dev/null <<'EOF'
[smtp.sendgrid.net]:587
apikey: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
EOF
```

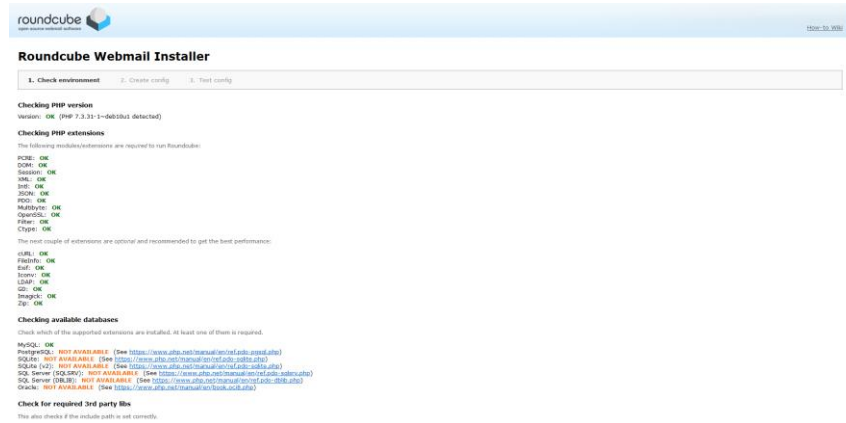
Y haremos que Postfix use ese fichero:

```
sudo chmod 600 /etc/postfix/sasl_passwd
sudo postmap /etc/postfix/sasl_passwd
```

Reiniciamos Postfix y Dovecot:

```
sudo systemctl restart postfix dovecot
```

Entramos en <https://mail.sysadminsolutions.cf/installer/> para finalizar la instalación de Roundcube:



En la segunda página tendremos que configurar los parámetros de nuestra base de datos creada para Roundcube y poner en el campo de host IMAP y SMTP la URI “tls://mail.sysadminsolutions.cf”, además instalaremos todos los plugins disponibles menos el de cambiar contraseña.

Ahora tendremos que copiar la configuración en /var/www/roundcube/config/, para ello la descargamos en el directorio /tmp y la movemos:

```
sudo mv /tmp/config.inc.php /var/www/roundcube/config/
```

Para que el dominio por defecto sea el correcto al introducir solo un usuario haremos:

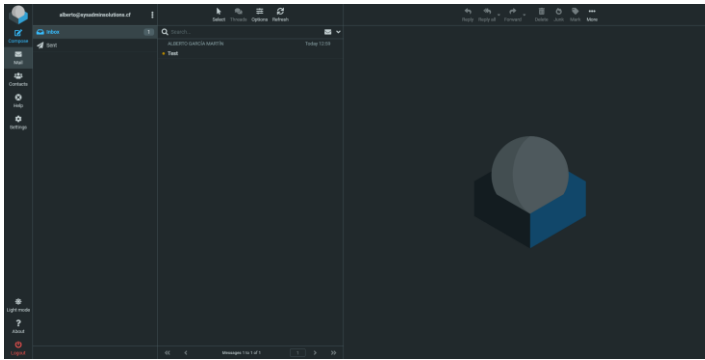
```
sudo tee -a /var/www/roundcube/config/config.inc.php > /dev/null
<<'EOF'
$config['mail_domain'] = '%d';
EOF
```

Por último, comprobaremos que nuestra configuración funciona y eliminaremos el directorio del instalador web:

```
sudo rm -rf /var/www/roundcube/installer/
```

Ya podemos acceder a Roundcube sin problemas:





Adicionalmente la práctica nos dice que los buzones de correo estarán limitados a 3MB por cada usuario, por lo que activaremos las cuotas en Dovecot y en los servicios de LMTP y IMAP:

```
sudo sed -i 's/#mail_plugins = /mail_plugins = quota/g'
/etc/dovecot/conf.d/10-mail.conf
sudo sed -i 's/#mail_plugins = $mail_plugins/mail_plugins =
$mail_plugins quota/g' /etc/dovecot/conf.d/20-lmtp.conf
sudo sed -i 's/#mail_plugins = $mail_plugins/mail_plugins =
$mail_plugins imap_quota/g' /etc/dovecot/conf.d/20-imap.conf
```

Ahora asignaremos el tamaño de la cuota (3MB), el límite para avisar al usuario editando el archivo correspondiente (95%) y el backend a usar:

```
sudo sed -i 's/#quota_rule = *:storage=1G/quota_rule = *:storage=3M/g'
/etc/dovecot/conf.d/90-quota.conf
sudo sed -i 's/#quota_warning = storage=95% quota-warning 95
%u/quota_warning = storage=95% quota-warning 95 %u/g'
/etc/dovecot/conf.d/90-quota.conf
sudo sed -i 's/#quota = maildir:User quota/quota = maildir:User
quota/g' /etc/dovecot/conf.d/90-quota.conf
```

Además, descomentaremos las líneas de service quota-warning para ejecutar un script cuando se alcance el límite para avisar al usuario. También cambiamos el usuario a mail:

```
# Example quota-warning service. The unix listener's permissions should be
# set in a way that mail processes can connect to it. Below example assumes
# that mail processes run as vmail user. If you use mode=0666, all system users
# can generate quota warnings to anyone.
service quota-warning {
    executable = script /usr/local/bin/quota-warning.sh
    user = dovecot
    unix_listener quota-warning {
        user = mail
    }
}
```

Crearemos un script que mande un correo en /usr/local/bin/quota-warning.sh:

```
sudo nano /usr/local/bin/quota-warning.sh :

#!/bin/bash
PERCENT=$1
USER=$2
```

```
cat << EOF | /usr/sbin/sendmail $USER -O "plugin/quota=maildir:User
quota:noonenforcing"
From: alberto@sysadminsolutions.cf
Subject: Aviso de cuota
```

```
Tu buzón de correo está al $PERCENT%
EOF
```

```
sudo chmod a+x /usr/local/bin/quota-warning.sh
```

Ahora reiniciaremos Dovecot:

```
sudo systemctl restart dovecot
```

Al ejecutar el comando “sudo doveadm quota get -A” podemos observar que la cuota para el buzón de correo está en funcionamiento:

Username	Quota name	Type	Value	Limit	%
nobody	User quota	STORAGE	error	error	error
nobody	User quota	MESSAGE	error	error	error
alberto	User quota	STORAGE	13	3072	0
alberto	User quota	MESSAGE	7	-	0
systemd-coredump	User quota	STORAGE	error	error	error
systemd-coredump	User quota	MESSAGE	error	error	error
prueba	User quota	STORAGE	2	3072	0
prueba	User quota	MESSAGE	1	-	0

Roundcube también nos mostrará la cuota disponible.

Ahora vinculamos Roundcube a LemonLDAP para que los usuarios puedan acceder a su correo directamente desde el portal sin tener que volver a introducir su contraseña, el proceso es análogo que el que seguimos para FusionDirectory. Pero primero tenemos que habilitar la opción de guardar la contraseña en la sesión para mandarla a Roundcube:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 set
storePassword 1
```

Añadiremos el dominio a los virtual hosts:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
addKey \
'locationRules/mail.sysadminsolutions.cf' 'default' 'accept' \
'locationRules/mail.sysadminsolutions.cf'
'(?#Logout)^\/*_task\=logout' 'logout_sso' \
'exportedHeaders/mail.sysadminsolutions.cf' 'Auth-User' '$uid' \
'exportedHeaders/mail.sysadminsolutions.cf' 'Auth-Pw'
'$_password'
```

Y modificaremos el archivo /etc/nginx/sites-available/mail.sysadminsolutions.cf reemplazando la sección de location \.php\$ existente por:

```
location = /lmauth {
    internal;
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/llng-fastcgi-server/llng-fastcgi.sock;
    fastcgi_pass_request_body off;
    fastcgi_param CONTENT_LENGTH "";
```

```

        fastcgi_param HOST $http_host;
        fastcgi_param X_ORIGINAL_URI $original_uri;
    }

    location ~* \.php$ {
        auth_request /lmauth;
        set $original_uri $uri$is_args$args;
        auth_request_set $lmremote_user $upstream_http_lm_remote_user;
        auth_request_set $lmlocation $upstream_http_location;
        auth_request_set $cookie_value $upstream_http_set_cookie;
        add_header Set-Cookie $cookie_value;
        error_page 401 $lmlocation;
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/run/php/php7.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        include fastcgi_params;
        include snippets/lmg-lua-headers.conf;
    }

```

**Reiniciamos Nginx:**

```
sudo nginx -t && sudo systemctl restart nginx
```

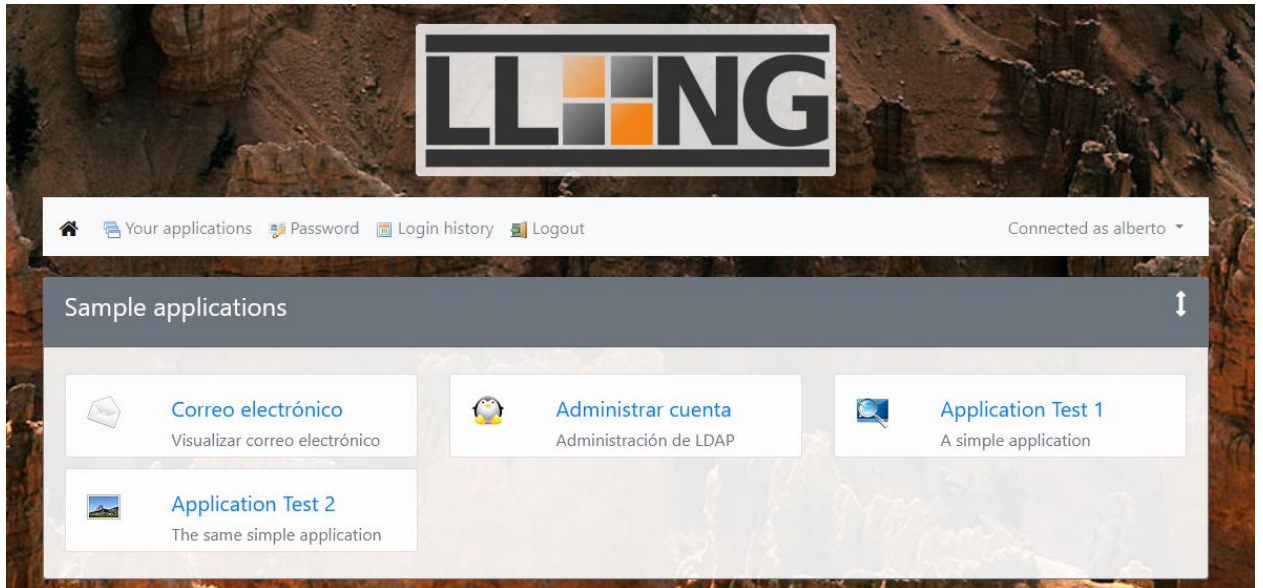
**Ahora tenemos que modificar el código del plugin http\_authentication de Roundcube para sustituir todas las instancias de PHP\_AUTH\_\* por HTTP\_AUTH\_\*:**

```
sudo sed -i 's/PHP_AUTH_/HTTP_AUTH_/g'
/var/www/roundcube/plugins/http_authentication/http_authentication.php
```

**Ahora añadiremos la aplicación de Roundcube al portal:**

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
    applicationList/1sample/email type application \
    applicationList/1sample/email/options description "Accede a
Roundcube" \
    applicationList/1sample/email/options display "on" \
    applicationList/1sample/email/options logo "mailappt.png" \
    applicationList/1sample/email/options name "Correo electrónico" \
    applicationList/1sample/email/options uri
"https://mail.sysadminsolutions.cf/"
```

**Ya podemos acceder al correo electrónico desde el portal sin volver a hacer login**



Finalmente, para que LemonLDAP pueda mandar correos a los usuarios estableceremos los parámetros del servidor SMTP:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
set \
SMTPAuthUser 'alberto' \
SMTPAuthPass 'password' \
SMTPServer 'mail.sysadminsolutions.cf' \
SMTPPort '587' \
SMTPTLS 'starttls' \
mailFrom 'noreply@sysadminsolutions.cf'
```

## 2.6. Web personal

Para que los usuarios puedan crear su propia web personal instalaremos la herramienta WordPress, para ello primero tenemos que crear una base de datos en MariaDB con un usuario para que WordPress pueda usarla:

```
sudo mariadb
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE
utf8_unicode_ci;
GRANT ALL ON wordpress.* TO 'wordpress_user'@'localhost' IDENTIFIED BY
'1kB5Gnv3GxZRiS';
FLUSH PRIVILEGES;
EXIT;
```

Ahora instalaremos las dependencias que requiere WordPress:

```
sudo apt install php-curl php-gd php-intl php-mbstring php-soap php-xml
php-xmlrpc php-zip
```

Nos descargamos la última versión de WordPress, la extraemos, la movemos dentro de /var/www y cambiamos el propietario a www-data:

```
curl -LO https://wordpress.org/latest.tar.gz
```

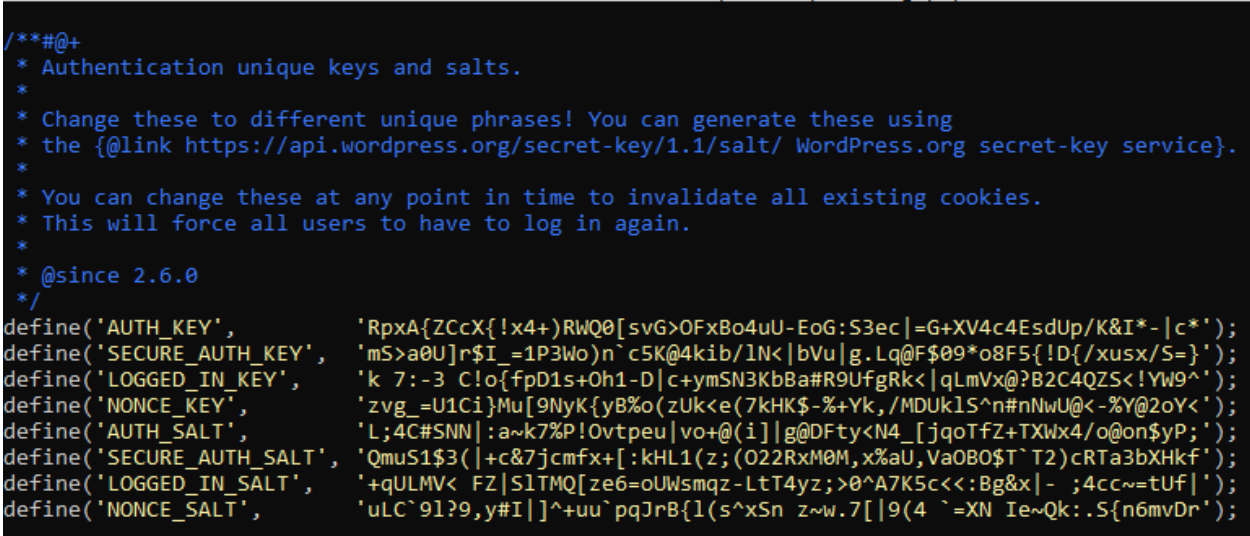
```
tar xzvf latest.tar.gz && rm latest.tar.gz
sudo mv wordpress /var/www/
sudo cp /var/www/wordpress/wp-config-sample.php /var/www/wordpress/wp-
config.php
sudo chown -R www-data:www-data /var/www/wordpress
```

Para mejorar la seguridad de nuestra instalación de WordPress tenemos que proporcionarle unas claves que generamos con la orden:

```
curl -s https://api.wordpress.org/secret-key/1.1/salt/
```

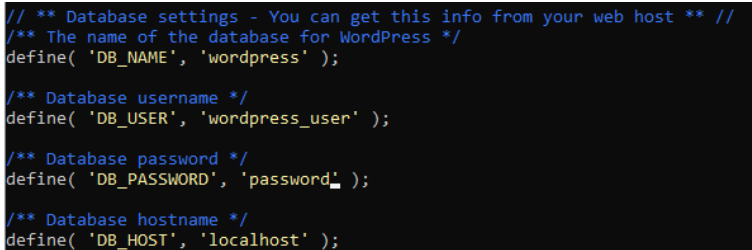
Ahora editaremos el archivo de configuración para incluir los valores generados:

```
sudo nano /var/www/wordpress/wp-config.php
```



```
/*#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',          'RpXA{ZCcX{!x4+)RWQ0[svG>OFxBo4uU-EoG:S3ec|=G+XV4c4EsdUp/K&I*-|c*');
define('SECURE_AUTH_KEY',  'mS>a0U]r$I _=1P3Wo)n`c5K@4kib/1N<|bVu|g.Lq@F$09*o8F5{!D{/xusx/S=}')';
define('LOGGED_IN_KEY',    'k 7:-3 C!o{fpD1s+0h1-D|c+ymSN3KbBa#R9UfgRk<|qLmVx@?B2C4QZS<!YW9^');
define('NONCE_KEY',       'zvg _=U1Ci}Mu[9NyK{yB%o(zUk<e(7kHK$-%+Yk,/MDUK1S^nnNwU@<-%Y@2oY<');
define('AUTH_SALT',       'L;4C#SNN|:a~k7%P!Ovtpeu|vo+@(i)|g@DFty<N4_[jqoTfZ+TXWx4/o@on$yP;');
define('SECURE_AUTH_SALT', 'QmuS1$3(|+c&7jcmfx+[:kHL1(z;(O22RxM0M,x%aU,Va0B0$T`T2)cRTa3bXHkf');
define('LOGGED_IN_SALT',  '+qULMV< FZ|S1TMQ[ze6=ouWsmqz-LtT4yz;>0^A7K5c<<:Bg&x|- ;4cc~=tUf|');
define('NONCE_SALT',      'uLC`9l?9,y#I|^+uu`pqJrB{1(s^xSn z~w.7[|9(4 `=XN Ie~Qk:.S{n6mvDr');
```

También en la configuración editaremos los parámetros de la base de datos creada para que WP pueda acceder a ella:



```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpress_user' );

/** Database password */
define( 'DB_PASSWORD', 'password_' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Y al final del archivo añadimos la línea “define('FS\_METHOD', 'direct');” para indicar que puede escribir directamente al disco.

Crearemos un bloque de Nginx para poder acceder a WordPress desde blog.sysadminsolutions.cf:

```
sudo tee /etc/nginx/sites-available/blog.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen 80;
```

```

root /var/www/wordpress;
index index.php;
server_name blog.sysadminsolutions.cf;

location = /favicon.ico { log_not_found off; access_log off; }
location = /robots.txt { log_not_found off; access_log off; allow
all; }
location ~* \.(css|gif|ico|jpeg|jpg|js|png)$ {
    expires max;
    log_not_found off;
}

location / {
    try_files $uri $uri/ /index.php$is_args$args;
}

location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
}
}
EOF

sudo ln -s /etc/nginx/sites-available/blog.sysadminsolutions.cf
/etc/nginx/sites-enabled/

```

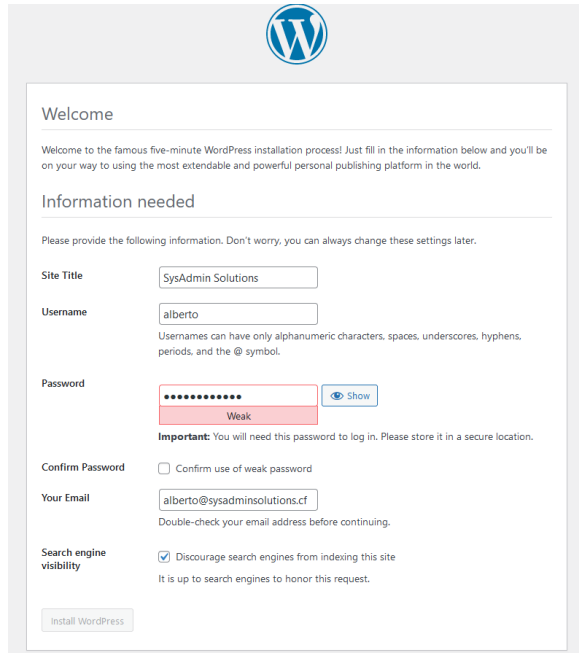
**Comprobamos la configuración y reiniciamos Nginx:**

```
sudo nginx -t && sudo systemctl reload nginx
```

**Y obtenemos un certificado SSL para blog.sysadminsolutions.cf:**

```
sudo certbot --redirect --nginx -d blog.sysadminsolutions.cf
```

Ahora seguiremos la instalación desde <https://blog.sysadminsolutions.cf/wp-admin/install.php>, donde introduciremos el nombre del sitio y el nombre, contraseña y correo del administrador:



The image shows the WordPress installation 'Welcome' screen. At the top is the WordPress logo. Below it, a 'Welcome' heading is followed by a brief introduction. The 'Information needed' section prompts the user to provide site details. The 'Site Title' field contains 'SysAdmin Solutions'. The 'Username' field contains 'alberto', with a note that usernames can only contain alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol. The 'Password' field shows a masked password with a 'Show' button; a red box below it indicates the password is 'Weak', and an 'Important' note states the password is needed for login. The 'Confirm Password' section has an unchecked checkbox for 'Confirm use of weak password'. The 'Your Email' field contains 'alberto@sysadminsolutions.cf', with a note to double-check the email. The 'Search engine visibility' section has a checked checkbox for 'Discourage search engines from indexing this site', with a note that it is up to search engines to honor the request. At the bottom is an 'Install WordPress' button.

Ahora tendremos que activar Multisite para que cada usuario pueda tener su propia instancia de WordPress, para ello añadiremos en `/var/www/wordpress/wp-config.php` la línea `"define('WP_ALLOW_MULTISITE', true );"`

Una vez editada la configuración podemos ir al panel de control de WordPress y acceder a Tools > Network Setup, aquí seleccionaremos la opción de usar sub-directorios y damos a Install, nos devolverá una serie de configuraciones que debemos añadir en el fichero de configuración.

```
sudo nano /var/www/wordpress/wp-config.php
```

```
GNU nano 3.2 /var/www/wordpress/wp-config.php
* Change this to true to enable the display of notices during development.
* It is strongly recommended that plugin and theme developers use WP_DEBUG
* in their development environments.
*
* For information on other constants that can be used for debugging,
* visit the documentation.
*
* @link https://wordpress.org/support/article/debugging-in-wordpress/
*/
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */
define( 'FS_METHOD', 'direct' );
define( 'WP_ALLOW_MULTISITE', true );
define( 'MULTISITE', true );
define( 'SUBDOMAIN_INSTALL', false );
define( 'DOMAIN_CURRENT_SITE', 'blog.sysadminsolutions.cf' );
define( 'PATH_CURRENT_SITE', '/' );
define( 'SITE_ID_CURRENT_SITE', 1 );
define( 'BLOG_ID_CURRENT_SITE', 1 );

/* That's all, stop editing! Happy publishing. */
```

También tendremos que editar el fichero de configuración de Nginx:

```
sudo tee /etc/nginx/sites-available/blog.sysadminsolutions.cf >
/dev/null <<'EOF'
```

```

server {
    listen 80;

    root /var/www/wordpress;
    index index.php;
    server_name blog.sysadminsolutions.cf;

    location = /favicon.ico { log_not_found off; access_log off; }
    location = /robots.txt { log_not_found off; access_log off; allow
all; }
    location ~* \.(css|gif|ico|jpeg|jpg|js|png)$ {
        expires max;
        log_not_found off;
    }

    location / {
        try_files $uri $uri/ /index.php$is_args$args;
    }
    if (!-e $request_filename) {
        rewrite /wp-admin$ $scheme://$host$uri/ permanent;
        rewrite ^(/[^\/]*)?(/wp-.*$) $2 last;
        rewrite ^(/[^\/]*)?(/*.*\.php) $2 last;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
    }
}
EOF

```

**Comprobamos la configuración y reiniciamos Nginx:**

```
sudo nginx -t && sudo systemctl reload nginx
```

**Tendremos que reinstalar el certificado para poder acceder a través de HTTPS, seleccionamos la opción 1 cuando nos pregunte:**

```
sudo certbot --redirect --nginx -d blog.sysadminsolutions.cf
```

**Ahora refrescamos la página y volvemos a introducir nuestro login, para permitir a los usuarios crear nuevas páginas activamos la opción correspondiente en Settings > Network Settings.**

Para poder vincular los usuarios de nuestro sistema a los de WordPress instalaremos el plugin Authorizer desde el portal de WordPress, después de instalarlo vamos a los ajustes y marcaremos las casillas para aplicar los ajustes a todos los sitios. Seleccionaremos también que todos los usuarios con cuenta puedan hacer login.

En la pestaña de External Service habilitaremos el login mediante CAS, en el hostname pondremos “sysadminsolutions.cf”, el puerto será el 443, el path será “/cas/”, el protocolo CAS 2.0, el atributo de email “@sysadminsolutions.cf”, marcaremos la casilla de “Automatic login” y en Advanced marcaremos la casilla de “Hide WordPress login”.



Para limitar el número de blogs que puede crear cada usuario descargamos e instalamos desde el portal de WordPress el siguiente plugin: <https://github.com/toddnestor/limit-blogs-per-user>, una vez instalado iremos a Network Settings y cambiaremos el número de blogs permitido por usuario a 1.

Ahora vinculamos WordPress a LemonLDAP para que los usuarios puedan acceder a la administración de su blog directamente desde el portal sin tener que volver a introducir su contraseña. Esta vez usaremos CAS para la autenticación por lo que agregaremos los parámetros necesarios:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        casAppMetaDataExportedVars/wordpress mail mail \
        casAppMetaDataExportedVars/wordpress givenName givenName \
        casAppMetaDataExportedVars/wordpress sn sn \
        casAppMetaDataOptions/wordpress casAppMetaDataOptionsService
'https://blog.sysadminsolutions.cf/' \
    casAppMetaDataOptions/wordpress
casAppMetaDataOptionsUserAttribute uid
```

Haremos la página raíz de WordPress privada para que los usuarios tengan que iniciar sesión para verla y cambiaremos el contenido para que muestre a los usuarios instrucciones de cómo crear un sitio web.



# ¡Crea tu web personal!

Haz click aquí

El límite de blogs por usuario es uno, si ya has creado tu sitio puedes administrarlo desde [aquí](#).

[SysAdmin Solutions](#)

Por último, añadiremos un enlace en el portal que lleve directamente al usuario a la página de administración de sitios de WordPress:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        applicationList/1sample/blog type application \
```

```

        applicationList/1sample/blog/options description "Administración
del blog" \
        applicationList/1sample/blog/options display "on" \
        applicationList/1sample/blog/options logo "web.png" \
        applicationList/1sample/blog/options name "Web personal" \
        applicationList/1sample/blog/options uri
"https://blog.sysadminsolutions.cf/wp-admin/my-sites.php"

```

Ya podemos acceder a WordPress desde el portal sin volver a hacer login.

## 2.7. SFTP

El servidor de SFTP es el mismo que el de SSH por lo que ya se encuentra instalado, y el acceso a este servicio queda logueado en `/var/log/auth.log` de forma automática.

Para que nuestros usuarios puedan acceder a sus archivos desde una interfaz web moderna instalaremos Filestash, que puede actuar como un cliente SFTP local, también habrá que instalar sus dependencias. Para ello usaremos las siguientes órdenes:

```

curl --resolve golang.org -sL https://golang.org/dl/$(curl
https://go.dev/VERSION?m=text).src.tar.gz | tar xzf - && sudo mv go
/usr/local
wget -O - https://pastebin.com/raw/gx46nVSS | sudo apt-key add -
curl --resolve downloads.filestash.app -s
https://downloads.filestash.app/latest/filestash_`uname -s`-`uname -
m`.tar.gpg | gpg --decrypt | tar xf -
sudo mv filestash /srv/app/
sudo apt install -y curl tor emacs-nox ffmpeg zip poppler-utils
sudo useradd filestash
sudo chown -R filestash:filestash /srv/app/
sudo tee /srv/app/filestash.service > /dev/null <<'EOF'
[Unit]
Description=Filestash, a modern web client for SFTP, S3, files,
WebDAV...
Documentation=https://www.filestash.app/docs/
Requires=network.target
After=network.target

[Service]
User=filestash
Group=filestash
Type=simple
ExecStart=/srv/app/filestash

[Install]
WantedBy=multi-user.target
EOF
sudo systemctl enable /srv/app/filestash.service
sudo systemctl restart filestash

```

Además, crearemos un bloque de Nginx para acceder desde `files.sysadminsolutions.cf`:

```

sudo tee /etc/nginx/sites-available/files.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen      80;
    server_name files.sysadminsolutions.cf;

    client_max_body_size 1024M;

    location / {
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Origin '';

        proxy_pass      http://127.0.0.1:8334;
        proxy_read_timeout 86400;
    }
}
EOF

```

**Activamos el sitio web:**

```

sudo ln -s /etc/nginx/sites-available/files.sysadminsolutions.cf
/etc/nginx/sites-enabled/files.sysadminsolutions.cf
sudo nginx -t && sudo systemctl restart nginx

```

**Y obtenemos un certificado SSL para el subdominio:**

```

sudo certbot --redirect --nginx -d files.sysadminsolutions.cf

```

Ahora podemos acceder a <https://files.sysadminsolutions.cf> y finalizar la instalación introduciendo una contraseña de administrador y seleccionando como Backend sólo SFTP e introducimos en Hostname “sysadminsolutions.cf”.

Los requisitos de la práctica indican que los usuarios no pueden navegar entre otros directorios fuera de su directorio personal, como por defecto ya estamos creando los directorios de los usuarios dentro de /home con una máscara que solo da permisos de lectura al propietario solo nos queda “enjaular” a nuestros usuarios dentro de /home.

En la configuración del servidor de SSH indicaremos que los usuarios del grupo alumnos y profesores solo puedan acceder a través de SFTP y que su directorio root sea /home:

```

sudo tee -a /etc/ssh/sshd_config > /dev/null <<'EOF'
Match Group alumnos,profesores
    ChrootDirectory /home
    AllowTCPForwarding no
    X11Forwarding no
    ForceCommand internal-sftp -d %u

```

EOF

Y reiniciamos el servidor SSH:

```
sudo systemctl restart sshd
```

Ahora vinculamos Filestash a LemonLDAP para que los usuarios puedan acceder a sus archivos directamente desde el portal sin tener que volver a introducir su contraseña.

Añadimos el dominio a los virtual hosts:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        'locationRules/files.sysadminsolutions.cf' 'default' 'accept' \
        'locationRules/files.sysadminsolutions.cf' '(?#Logout)^/logout'
'logout_sso'
```

La única forma que tiene Filestash de recibir los parámetros de login es a través de parámetros GET por lo que activaremos en LemonLDAP el proveedor de GET e indicamos los parámetros a pasar:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 set
issuerDBGetActivation 1
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        'macros' '_type' "'sftp'" \
        'macros' '_hostname' "'sysadminsolutions.cf'" \
        'macros' '_password' "\"\$_password\"" \
        'issuerDBGetParameters/files.sysadminsolutions.cf' 'username'
'uid' \
        'issuerDBGetParameters/files.sysadminsolutions.cf' 'password'
'_password' \
        'issuerDBGetParameters/files.sysadminsolutions.cf' 'type' '_type'
\
        'issuerDBGetParameters/files.sysadminsolutions.cf' 'hostname'
'_hostname'
```

Para pasarle la URL de login a LemonLDAP la tenemos que codificar en base64:

`https://files.sysadminsolutions.cf/login =>`

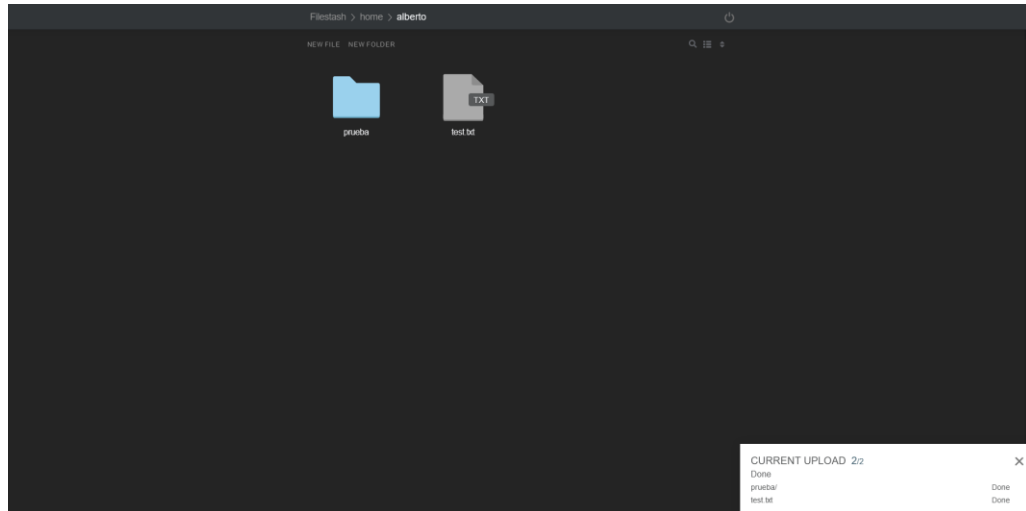
`aHR0cHM6Ly9maWxlcY5zeXNhZG1pbmNvbHV0aW9ucy5jZi9sb2dpbg==`

Ahora añadiremos la aplicación de Filestash al portal, pasaremos la URL de LemonLDAP para que haga de pasarela y el usuario no tenga que volver a introducir su login:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        applicationList/1sample/files type application \
        applicationList/1sample/files/options description "Conéctate a
SFTP" \
        applicationList/1sample/files/options display "on" \
        applicationList/1sample/files/options logo "folder.png" \
        applicationList/1sample/files/options name "Explorador de
archivos" \
```

```
applicationList/1sample/files/options uri
"https://sysadminsolutions.cf/get/login?url=aHR0cHM6Ly9maWxlcy5zeXNhZG1
pbnNvbHV0aW9ucy5jZi9sb2dpcg=="
```

Ya podemos acceder al explorador de archivos desde el portal sin volver a hacer login.



## 2.8. Moodle

Procederemos a instalar el programa Moodle para nuestros usuarios, primero descargamos y extraemos los archivos en el directorio `/var/www/`:

```
wget https://download.moodle.org/download.php/direct/stable400/moodle-
latest-400.tgz
sudo tar xf moodle-latest-400.tgz -C /var/www/
rm moodle-latest-400.tgz
```

Crearemos una carpeta de datos de Moodle donde pueda escribir el servidor web y le damos permisos adecuados:

```
sudo mkdir /var/www/moodledata
sudo chown www-data:www-data /var/www/moodledata/
sudo chown -R www-data:www-data /var/www/moodle
```

Ahora crearemos la base de datos que usará Moodle:

```
sudo mysql
CREATE DATABASE moodle CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
GRANT ALL PRIVILEGES ON moodle.* TO 'moodle_user'@'localhost'
IDENTIFIED BY 'pAxvpJJcRVC47P';
FLUSH PRIVILEGES;
quit;
```

Para evitar problemas vamos a cambiar la variable `max_input_vars` de PHP a 5000:

```
sudo sed -i 's/max_input_vars = 1000/max_input_vars = 5000/g'
/etc/php/7.3/fpm/php.ini
```

Ahora añadiremos un bloque de Nginx para acceder a la página de Moodle desde moodle.sysadminsolutions.cf:

```
sudo tee /etc/nginx/sites-available/moodle.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen 80;
    root /var/www/moodle;
    index index.php index.html index.htm;
    server_name moodle.sysadminsolutions.cf;

    location ~ [^/]\.php(/|$) {
        fastcgi_split_path_info ^(.+\.(php))(/.+)$;
        fastcgi_index        index.php;
        fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
        include               fastcgi_params;
        fastcgi_param        PATH_INFO        $fastcgi_path_info;
        fastcgi_param        SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    }

    location / {
        try_files $uri $uri/ =404;
    }
}
EOF

sudo ln -s /etc/nginx/sites-available/moodle.sysadminsolutions.cf
/etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl restart nginx
```

Y obtenemos un certificado SSL para el subdominio:

```
sudo certbot --redirect --nginx -d moodle.sysadminsolutions.cf
```

También crearemos una tarea en cron que se ejecute cada minuto para realizar diversas tareas que requiere Moodle:

```
sudo crontab -u www-data -e
* * * * * /usr/bin/php /var/www/moodle/admin/cli/cron.php >/dev/null
```

Ahora podemos ir a <https://moodle.sysadminsolutions.cf> para finalizar la instalación, cuando nos pregunte por el tipo de base de datos seleccionamos MariaDB e introducimos los datos de la base de datos creada anteriormente. Posteriormente introduciremos los datos de nuestro usuario administrador, luego rellenamos los campos del sitio y finalizamos la instalación.

Para permitir que los usuarios ya registrados en nuestro servidor accedan directamente a Moodle habilitaremos el plugin CAS yendo a Site administration > Plugins > Authentication > Manage authentication y activando el plugin de CAS (SSO). En hostname introducimos "sysadminsolutions.cf", en base URI "/cas/", el puerto 443 y marcamos la opción de CAS logout.

Luego en los ajustes del servidor de LDAP introducimos “ldap://sysadminsolutions.cf” y marcamos el uso de TLS, en distinguished name introducimos “cn=admin,dc=sysadminsolutions,dc=cf” y la contraseña, en contexto introducimos “ou=people,dc=sysadminsolutions,dc=cf” y en user attribute “uid”.

Ahora tenemos que configurar LemonLDAP para que le envíe a Moodle los parámetros de autenticación:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        casAppMetaDataExportedVars/moodle mail mail \
        casAppMetaDataExportedVars/moodle givenName givenName \
        casAppMetaDataExportedVars/moodle sn sn \
        casAppMetaDataOptions/moodle casAppMetaDataOptionsService
'https://moodle.sysadminsolutions.cf/' \
        casAppMetaDataOptions/moodle casAppMetaDataOptionsUserAttribute
uid
```

Para volver a permitir al usuario administrador entrar sin volver a hacer login tenemos que editar su entrada en la base de datos de Moodle:

```
sudo mysql
CONNECT moodle;
UPDATE mdl_user SET auth='cas' where username='alberto';
exit;
```

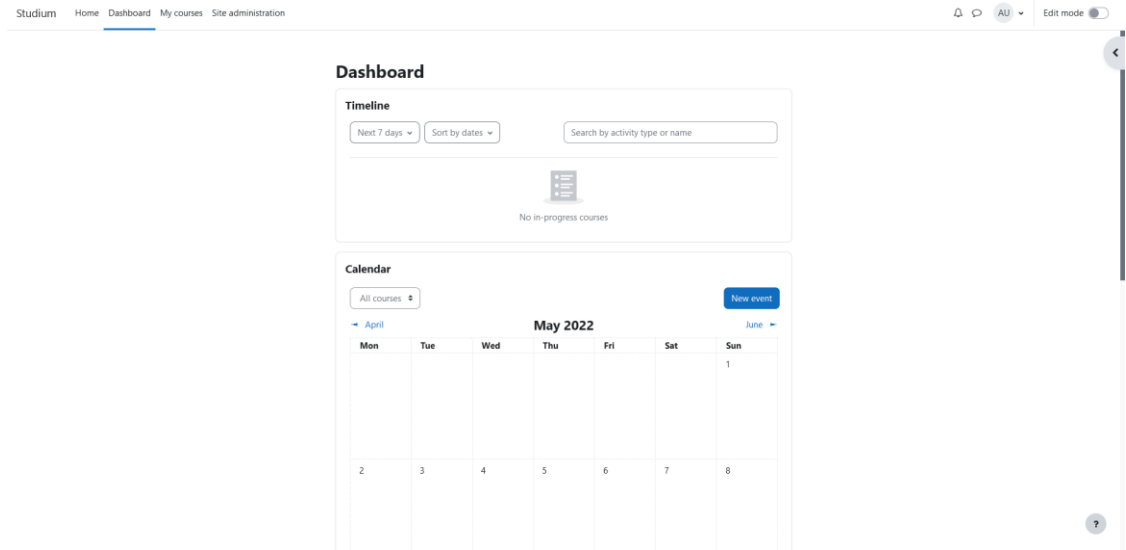
A continuación, haremos algunos ajustes finales a la instalación, en General>Security>Site security settings, habilitaremos la opción de “Force users to log in” para mostrar la pantalla de login nada más entrar al sitio. En Server>Email configuraremos los parámetros de nuestro servidor de correo, SMTP host “mail.sysadminsolutions.cf:587”, en security “TLS” y nuestro login.

En Server>System paths rellenamos los campos para que Moodle pueda acceder a diversos binarios.

Ahora añadiremos un enlace en el portal que lleve directamente al usuario a la página de Moodle:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        applicationList/1sample/moodle type application \
        applicationList/1sample/moodle/options description "Accede a
Moodle" \
        applicationList/1sample/moodle/options display "on" \
        applicationList/1sample/moodle/options logo "demo.png" \
        applicationList/1sample/moodle/options name "Stadium" \
        applicationList/1sample/moodle/options uri
'https://moodle.sysadminsolutions.cf'
```

Ya podemos acceder a Moodle desde el portal sin volver a hacer login.



## 2.9. Monitorización

Para recolectar la información del sistema y visualizarla de una forma gráfica y sencilla instalaremos la herramienta Zabbix, primero instalaremos su repositorio:

```
wget https://repo.zabbix.com/zabbix/6.0/debian/pool/main/z/zabbix-release/zabbix-release_6.0-1+debian10_all.deb
sudo dpkg -i zabbix-release_6.0-1+debian10_all.deb
sudo apt update
rm zabbix-release_6.0-1+debian10_all.deb
```

Instalamos los paquetes correspondientes:

```
sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-nginx-conf zabbix-sql-scripts zabbix-agent
```

Creamos una base de datos para Zabbix, con un usuario para que pueda acceder a ella:

```
sudo mysql
create database zabbix character set utf8mb4 collate utf8mb4_bin;
create user zabbix@localhost identified by 'RzE5wUfPeGOU2k';
grant all privileges on zabbix.* to zabbix@localhost;
quit;
```

Importamos a la base de datos los esquemas y datos iniciales, cuando nos pida la contraseña introducimos la que acabamos de meter en la base de datos:

```
sudo zcat /usr/share/doc/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p zabbix
```

En la configuración de Zabbix añadimos la contraseña de la base de datos:

```
sudo sed -i 's/# DBPassword=/DBPassword=RzE5wUfPeGOU2k/g'
/etc/zabbix/zabbix_server.conf
```



En la configuración de Nginx de Zabbix indicamos el puerto 80 y el nombre de servidor “stats.sysadminsolutions.cf”:

```
sudo sed -i 's/# listen 80;/ listen 80;/g' /etc/zabbix/nginx.conf
sudo sed -i 's/# server_name example.com;/ server_name stats.sysadminsolutions.cf;/g' /etc/zabbix/nginx.conf
```

Por último, reiniciamos los servicios de Zabbix y los habilitamos para posteriores arranques:

```
sudo systemctl restart zabbix-server zabbix-agent nginx php7.3-fpm
sudo systemctl enable zabbix-server zabbix-agent nginx php7.3-fpm
```

Obtenemos un certificado SSL para el subdominio:

```
sudo certbot --redirect --nginx -d stats.sysadminsolutions.cf
```

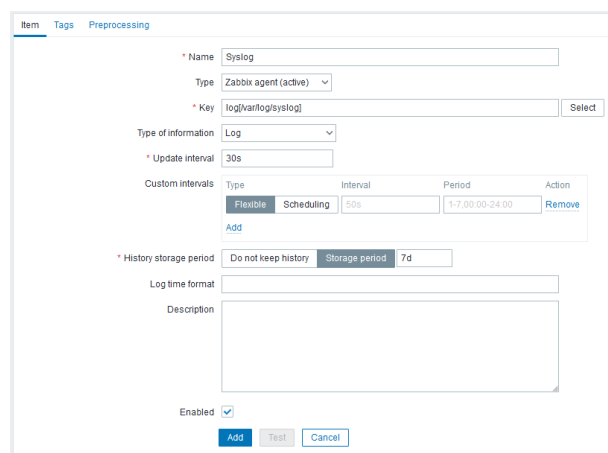
Ahora podemos seguir con la instalación en <https://stats.sysadminsolutions.cf>, cuando nos pregunte por la contraseña de la base de datos introducimos la anterior. En server name ponemos “sysadminsolutions.cf” y seleccionamos la zona horaria apropiada. El login por defecto es Admin/zabbix. Lo primero que haremos una vez dentro es cambiar las credenciales del Admin por las nuestras propias. En Configuration>Templates buscaremos las templates de Linux y las aplicaremos.

Ya podemos añadir widgets en la Dashboard que representen gráficas de uso de CPU, memoria, disco, etc.

Para que Zabbix pueda leer registros del sistema como syslog tenemos que añadir su usuario al grupo adm:

```
sudo usermod -a -G adm zabbix
```

Añadiremos un control de registros del sistema añadiendo un ítem con los siguientes valores:

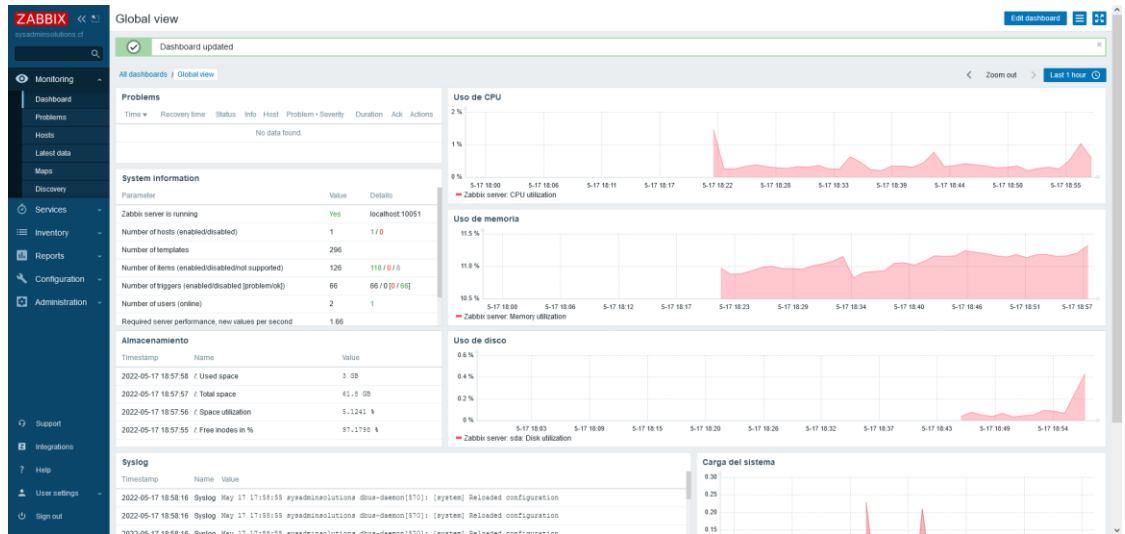


The screenshot shows the Zabbix web interface for configuring a new item. The 'Item' tab is selected. The configuration is as follows:

- Name: Syslog
- Type: Zabbix agent (active)
- Key: log[/var/log/syslog] (with a 'Select' button)
- Type of information: Log
- Update interval: 30s
- Custom intervals table:

Type	Interval	Period	Action
Flexible	Scheduling	50s	1:7:00:00-24:00
- History storage period: Do not keep history (Storage period: 7d)
- Log time format: (empty field)
- Description: (empty text area)
- Enabled: ☒
- Buttons: Add, Test, Cancel

Este es el resultado final de la página principal de monitorización:



Con el fin de recibir reportes diarios del sistema configuraremos antes el web service de Zabbix, para ello necesitaremos instalar también Google Chrome:

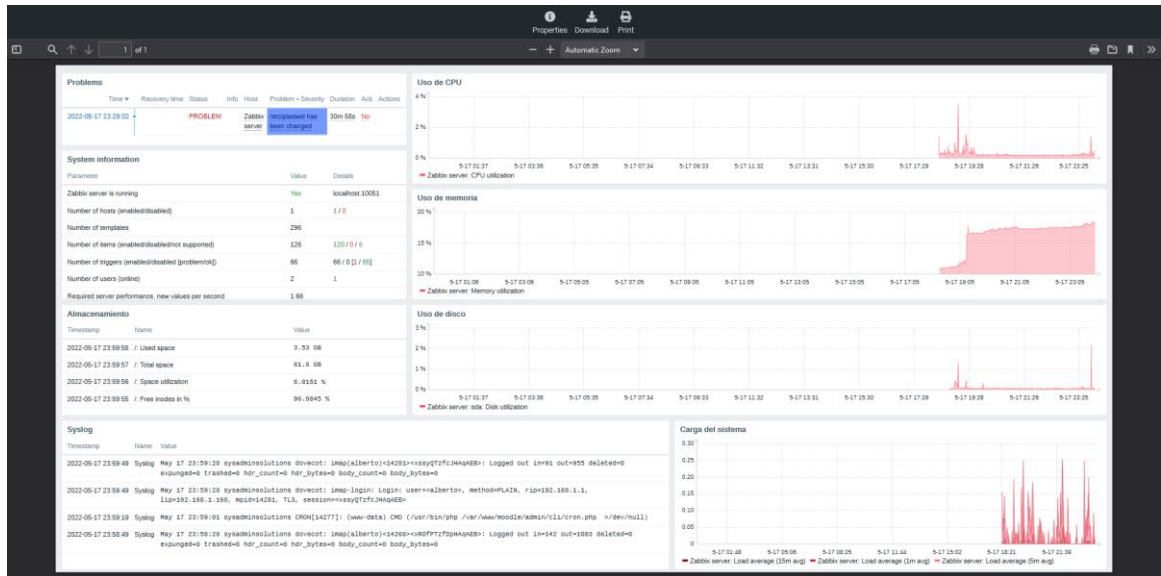
```
sudo apt install zabbix-web-service
wget https://dl.google.com/linux/direct/google-chrome-
stable_current_amd64.deb
sudo apt install ./google-chrome-stable_current_amd64.deb
rm google-chrome-stable_current_amd64.deb
```

Configuramos Zabbix para activar el servicio de reportes:

```
sudo sed -i 's/#
WebServiceURL=/WebServiceURL=http://localhost:10053/report/g'
/etc/zabbix/zabbix_server.conf
sudo tee -a /etc/zabbix/zabbix_server.conf > /dev/null <<'EOF'
StartReportWriters=3
EOF'
```

En Administration>General>Other, en el campo de Frontend URL introducimos "<https://stats.sysadminsolutions.cf/>". En Administration>Media types>Email configuraremos los parámetros de nuestro servidor de correo, server "mail.sysadminsolutions.cf", puerto 587, seguridad "StartTLS", helo "sysadminsolutions.cf", email "zabbix@sysadminsolutions.cf" y las credenciales de un usuario que pueda enviar correo. En User>Profile>Media añadiremos el correo donde queremos recibir el reporte "alberto@sysadminsolutions.cf".

Para crear una tarea que envíe el informe iremos a Reports>Scheduled reports y damos a Create report. Por defecto manda el informe al correo del usuario indicado todos los días a las 00:00 y la información del día anterior, por lo que solo tendremos que darle un nombre y rellenar los campos de Subject y Message.



Para añadir Zabbix a la lista de aplicaciones del portal y que solo le aparezca al administrador usaremos la siguiente orden:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
  applicationList/2administration/stats type application \
  applicationList/2administration/stats/options description
"Visualiza estadísticas" \
  applicationList/2administration/stats/options display '$uid =~
/^alberto$/ ' \
  applicationList/2administration/stats/options logo "bell.png" \
  applicationList/2administration/stats/options name "Monitor del
sistema" \
  applicationList/2administration/stats/options uri
"https://stats.sysadminsolutions.cf/"
```

## 2.10. Control de cuotas

Los usuarios estarán limitados a usar 80MB por lo que estableceremos un sistema de cuotas para controlar su uso de disco. Primero instalamos el paquete quota:

```
sudo apt install quota
```

Editamos fstab para añadir las opciones usrquota y grpquota

```
sudo sed -i 's/errors=remount-ro/errors=remount-ro,usrquota,grpquota/g'
/etc/fstab
```

Remontamos el sistema de archivos para que los cambios tomen efecto:

```
sudo mount -o remount /
```

Y activamos las cuotas con quotacheck:

```
sudo quotacheck -ugm /
```

Para que los usuarios tengan configuradas sus cuotas cuando se registren instalaremos el módulo `pam_setquota`, para ello descargaremos y compilaremos el código fuente:

```
sudo apt install git libpam0g-dev
git clone https://github.com/amirsafiallah/pam_setquota
cd pam_setquota
make
sudo make install
cd .. && rm -rf pam_setquota
```

Una vez instalado, editamos el fichero `/etc/pam.d/common-session` y añadimos la línea que activa las cuotas de 80MB en los usuarios, cuando alcancen el 95% se les avisará de que queda poco para que alcancen la cuota:

```
sudo tee -a /etc/pam.d/common-session > /dev/null <<'EOF'
session required /lib/security/pam_setquota.so bsoftlimit=77824
bhardlimit=81920 fs=/ startuid=1001
EOF
```

## 2.11. Web de status

Para mostrar la información de estatus de los servicios del sistema en una página web hemos creado un script que obtiene la salida de `systemctl` listando todos los procesos activos o inactivos y lo imprime en el archivo `/var/www/status/index.html`. Para estilizar el HTML pondremos usaremos un CSS.

```
sudo mkdir -p /var/www/status/html
sudo mv status.sh /var/www/status
sudo chmod u+x /var/www/status/status.sh
sudo mv status.css /var/www/status/html
sudo touch /var/www/status/html/index.html
sudo chown -R www-data:www-data /var/www/status
```

Hemos añadido al crontab de `www-data` el script `status.sh` para que se ejecute cada minuto:

```
sudo crontab -u www-data -e
* * * * * /usr/bin/bash /var/www/status/status.sh
```

Ahora crearemos el bloque Nginx para acceder a la página desde `status.sysadminsolutions.cf`:

```
sudo tee /etc/nginx/sites-available/status.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen 80;
    server_name status.sysadminsolutions.cf;

    root /var/www/status/html;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

EOF

```
sudo ln -s /etc/nginx/sites-available/status.sysadminsolutions.cf  
/etc/nginx/sites-enabled/  
sudo nginx -t && sudo systemctl restart nginx
```

**Y obtenemos un certificado SSL para el subdominio:**

```
sudo certbot --redirect --nginx -d status.sysadminsolutions.cf
```

Ya podemos acceder a la web de status:

SERVICIO	ESTADO	SUBESTADO	DESCRIPCION
backuppc.service	active	running	LSB: Launch backuppc server
cron.service	active	running	Regular background program processing daemon
dovecot.service	active	running	Dovecot IMAP/POP3 email server
filestash.service	active	running	Filestash, a modern web client for SFTP, S3, FTP, WebDAV...
lemonldap-ng-fastcgi-server.service	active	running	FastCGI server for Lemonldap:NG webssso system
mariadb.service	active	running	MariaDB 10.6.7 database server
nginx.service	active	running	A high performance web server and a reverse proxy server
php7.3-fpm.service	active	running	The PHP 7.3 FastCGI Process Manager
postfix.service	active	exited	Postfix Mail Transport Agent
postfix@-.service	active	running	Postfix Mail Transport Agent (instance -)
slapd.service	active	running	LSB: OpenLDAP standalone server (Lightweight Directory Access Protocol)
ssh.service	active	running	OpenBSD Secure Shell server
zabbix-agent.service	active	running	Zabbix Agent
zabbix-server.service	active	running	Zabbix Server
zabbix-web.service	active	running	Zabbix Web Service

Por último, añadiremos la web de status a la lista de apps del portal:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \  
    addKey \  
    applicationList/3documentation/status type application \  
    applicationList/3documentation/status/options description  
"Consulta el estado de los servicios" \  
    applicationList/3documentation/status/options display "on" \  
    applicationList/3documentation/status/options logo "bell.png" \  
    applicationList/3documentation/status/options name "Web de  
status" \  
    applicationList/3documentation/status/options uri  
"https://status.sysadminsolutions.cf/"
```

## 2.12. Copias de seguridad

Para realizar y mantener las copias de seguridad del sistema utilizaremos la herramienta BackupPC que permite realizar copias de seguridad con rsync ofreciendo una interfaz web. Primero descargamos e instalamos los paquetes:

```
wget https://backuppc4.s3-eu-west-1.amazonaws.com/Debian10/backuppc_4.4.0_amd64.deb  
wget https://backuppc4.s3-eu-west-1.amazonaws.com/Debian10/rsync-  
bpc_3.0.9.15_amd64.deb  
wget https://backuppc4.s3-eu-west-1.amazonaws.com/Debian10/libbackuppc-  
xs-perl_0.62-1_amd64.deb  
sudo apt install ./libbackuppc-xs-perl_0.62-1_amd64.deb  
sudo apt install ./rsync-bpc_3.0.9.15_amd64.deb  
sudo apt install ./backuppc_4.4.0_amd64.deb  
sudo cpanm SCGI  
rm -rf *.deb
```

Cuando el instalador nos pregunte si deseamos configurar BackupPC para Apache2 seleccionamos que no, ya que usaremos Nginx. Luego podemos iniciar el servicio de BackupPC:

```
sudo systemctl restart backuppc
```

Ahora crearemos un bloque de Nginx para acceder a BackupPC desde la dirección `backup.sysadminsolutions.cf`:

```
sudo tee /etc/nginx/sites-available/backup.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen 80;
    server_name backup.sysadminsolutions.cf;

    root /usr/share/backuppc/cgi-bin;
    index index.cgi;

    access_log /var/log/nginx/backuppc.access.log;
    error_log /var/log/nginx/backuppc.error.log;

    location / {
        location /backuppc {
            alias /usr/share/backuppc;
        }

        location ~ /\.cgi$ {
            include scgi_params;
            scgi_param REMOTE_USER $remote_user;
            scgi_param SCRIPT_NAME $document_uri;
            scgi_pass 127.0.0.1:10268;
        }
    }
}
EOF
```

```
sudo ln -s /etc/nginx/sites-available/backup.sysadminsolutions.cf
/etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl restart nginx
```

Y obtenemos un certificado SSL para el subdominio:

```
sudo certbot --redirect --nginx -d backup.sysadminsolutions.cf
```

Para configurar BackupPC, primero indicaremos el nombre de nuestro equipo en el archivo `hosts` para que haga la copia de seguridad:

```
sudo tee -a /etc/backuppc/hosts > /dev/null <<'EOF'
sysadminsolutions.cf 0 alberto
EOF
```

Indicaremos que use `rsync` para hacer las copias de seguridad:

```
sudo sed -i 's/${Conf}{XferMethod} = 'rsyncd';/${Conf}{XferMethod} =
'rsync';/g' /etc/backuppc/config.pl
sudo sed -i 's/${Conf}{RsyncBackupPCPath} = "";/${Conf}{RsyncBackupPCPath}
= "\usr\bin\rsync_bpc";/g' /etc/backuppc/config.pl
sed -i '/RsyncSshArgs/s/-l root/-l backuppc/' /etc/backuppc/config.pl
```

Por defecto el programa realiza una copia de seguridad completa cada 7 días y una incremental cada día, manteniendo copias de los últimos 7 días.

Configuraremos los directorios de los que queremos hacer copias de seguridad, excluyendo el propio directorio donde se guardan las copias:

```
sudo nano /etc/backuppc/config.pl
$Conf{BackupFilesExclude} = ['/var/lib/backuppc/'];
$Conf{BackupFilesOnly} = ['/home', '/etc', '/var'];
```

Para que BackupPC pueda conectarse por rsync para hacer las copias de seguridad tendremos que añadir sus claves SSH:

```
sudo su - backuppc
ssh-keygen
ssh-copy-id backuppc@sysadminsolutions.cf
```

Además, daremos permisos a backuppc para que pueda ejecutar rsync con sudo:

```
echo "backuppc    ALL=NOPASSWD:    $(which rsync) " | sudo tee
/etc/sudoers.d/backuppc
sudo nano /etc/backuppc/config.pl
$Conf{RsyncClientPath} = 'sudo /usr/bin/rsync';
```

Ahora entramos en la página <https://backup.sysadminsolutions.cf/> para realizar la primera copia de seguridad:

The screenshot shows the BackupPC web interface. On the left is a sidebar with navigation links: 'sysadminsolutions.cf Home', 'Browse backups', 'LOG file', 'LOG files', 'Edit Config', 'Hosts', 'Server', 'Status', 'Host Summary', 'Edit Config', 'Edit Hosts', 'Admin Options', 'LOG file', 'Old LOGs', 'Email summary', 'Current queues', 'Documentation', 'Wiki', and 'Homepage'. The main content area is titled 'Host sysadminsolutions.cf Backup Summary'. It includes a list of backup actions (Start Incr Backup, Start Full Backup, Stop/Dequeue Backup) and a 'Backup Summary' table. Below this is an 'Xfer Error Summary' table, a 'File Size/Count Reuse Summary' table, and a 'Compression Summary' table. The 'Backup Summary' table shows one backup (Backup# 0) that is active, with a level of 0, started on 2022-05-22 11:14, and has a duration of 0.0 minutes. The 'Xfer Error Summary' table shows no errors. The 'File Size/Count Reuse Summary' table shows no files. The 'Compression Summary' table shows no files.

Backup#	Type	Filled	Level	Start Date	Duration/mins	Age/days	Keep	Comment
0	active	yes	0	2022-05-22 11:14		0.0		Delete

Backup#	Type	View	#Xfer errs	#bad files	#bad share	#tar errs
0	active	XferLOG, Errors				

Backup#	Type	Totals	Existing Files	New Files				
		#files	Size/MiB	MiB/sec	#files	Size/MiB	#files	Size/MiB
0	active							

Backup#	Type	Comp Level	Existing Files	New Files				
			Size/MiB	Comp/MiB	Comp	Size/MiB	Comp/MiB	Comp
0	active	3						

Para impedir que usuarios normales accedan a esta página bloquearemos el acceso con LemonLDAP. Primero añadimos el dominio a los virtual hosts:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
addKey \
```



```
'locationRules/backup.sysadminsolutions.cf' 'default' '$uid =~
/^alberto$/'
```

Y modificaremos el bloque de Nginx reemplazando la sección de location `\.cgi$` existente por:

```
location = /lmauth {
    internal;
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/llng-fastcgi-server/llng-
fastcgi.sock;
    fastcgi_pass_request_body off;
    fastcgi_param CONTENT_LENGTH "";
    fastcgi_param HOST $http_host;
    fastcgi_param X_ORIGINAL_URI $original_uri;
}

location ~ /\.cgi$ {
    auth_request /lmauth;
    set $original_uri $uri$is_args$args;
    auth_request_set $lmremote_user
$upstream_http_lm_remote_user;
    auth_request_set $lmlocation $upstream_http_location;
    auth_request_set $cookie_value $upstream_http_set_cookie;
    add_header Set-Cookie $cookie_value;
    error_page 401 $lmlocation;
    include scgi_params;
    scgi_param REMOTE_USER $remote_user;
    scgi_param SCRIPT_NAME $document_uri;
    scgi_pass 127.0.0.1:10268;
}
```

Reiniciamos Nginx:

```
sudo nginx -t && sudo systemctl restart nginx
```

Para añadir BackupPC a la lista de apps del portal y que solo le aparezca al administrador usaremos la siguiente orden:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
    applicationList/2administration/backup type application \
    applicationList/2administration/backup/options description
"Administra BackupPC" \
    applicationList/2administration/backup/options display 'auto' \
    applicationList/2administration/backup/options logo "attach.png"
\
    applicationList/2administration/backup/options name "Copias de
seguridad" \
    applicationList/2administration/backup/options uri
"https://backup.sysadminsolutions.cf/"
```

## 2.13. Otros requisitos

### 2.13.1. Archivo condiciones.txt

Para crear el archivo “condiciones.txt” en la carpeta home de los usuarios cuando se registren, crearemos en el directorio /etc/skel un archivo con ese nombre:

```
sudo tee /etc/skel/condiciones.txt > /dev/null <<'EOF'  
Haz un uso responsable del sistema.  
EOF
```

### 2.13.2. Carpeta apuntes

Como los usuarios del sistema no pueden salir del directorio /home/ por los requisitos de la configuración de SFTP, hemos decidido crear un usuario de sistema llamado “apuntes” y usar la carpeta /home/apuntes como la carpeta compartida.

Creamos el usuario de sistema en Linux:

```
sudo adduser --system apuntes
```

Para deshabilitar el login a esa cuenta la bloquearemos:

```
sudo passwd -l apuntes
```

Ahora le asignamos al directorio /home/apuntes el propietario root y el grupo profesores, con permisos de lectura, escritura y ejecución para el grupo y con solo lectura y escritura para el resto.

```
sudo chown root:profesores /home/apuntes  
sudo chmod g+w /home/apuntes
```

Además, para evitar que los profesores borren archivos de otros profesores pondremos el sticky bit en la carpeta apuntes:

```
sudo chmod +t /home/apuntes
```

Para que el directorio aparezca en la carpeta home de los usuarios haremos un symlink de /home/apuntes en /etc/skel/, pero como los usuarios de SFTP creen que el root (/) es /home haremos que el symlink apunte a /apuntes:

```
sudo ln -s /apuntes /etc/skel/
```

### 2.13.3. Página de ayuda

Para la página de ayuda hemos modificado una plantilla HTML que hemos situado en /var/www/help/:

```
sudo mv help/ /var/www/  
sudo chown -R www-data:www-data /var/www/help/
```

Creamos un bloque de Nginx para acceder desde help.sysadminsolutions.cf:

```

sudo tee /etc/nginx/sites-available/help.sysadminsolutions.cf >
/dev/null <<'EOF'
server {
    listen 80;
    server_name help.sysadminsolutions.cf;

    root /var/www/help;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
EOF

sudo ln -s /etc/nginx/sites-available/help.sysadminsolutions.cf
/etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl restart nginx

```

**Y obtenemos un certificado SSL para el subdominio:**

```
sudo certbot --redirect --nginx -d help.sysadminsolutions.cf
```

**Por último, añadiremos la web de ayuda a la lista de apps del portal:**

```

sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
    addKey \
        applicationList/3documentation/help type application \
        applicationList/3documentation/help/options description
"Web con preguntas frecuentes" \
        applicationList/3documentation/help/options display "on" \
        applicationList/3documentation/help/options logo "help.png"
\
        applicationList/3documentation/help/options name "Ayuda" \
        applicationList/3documentation/help/options uri
"https://help.sysadminsolutions.cf/"

```



#### 2.13.4. Prevención de fork bombs

Para prevenir fork-bombs establecemos un límite de número máximo de procesos para todos los usuarios menos para el root:

```
sudo tee -a /etc/security/limits.conf > /dev/null <<'EOF'
*                hard  nproc          2048
root            hard  nproc          65536
EOF
```

### 2.14. Requisitos opcionales

#### 2.14.1. Aviso login administrador

Para mandar un correo electrónico al administrador, ejecutaremos un script Bash mediante pam\_exec cada vez que inicie sesión. Primero tenemos que instalar mailutils para mandar el correo:

```
sudo apt install mailutils
```

Luego creamos el script que manda el correo al administrador en caso de login en su cuenta o en la de root y añadimos la línea de pam\_exec en el fichero de common-session y common-session-noninteractive:

```
sudo tee /root/login-mail.sh > /dev/null <<'EOF'
#!/bin/sh
[ "$PAM_TYPE" = "open_session" ] || exit 0
if [ "${PAM_USER}" = "alberto" ] || [ "${PAM_USER}" = "root" ];
then
{
    echo "User: $PAM_USER"
    echo "Ruser: $PAM_RUSER"
    echo "Rhost: $PAM_RHOST"
    echo "Service: $PAM_SERVICE"
    echo "TTY: $PAM_TTY"
    echo "Date: $(date)"
    echo "Server: $(uname -a)"
} | mail -s "$(hostname -s) $PAM_SERVICE login: $PAM_USER"
'albertogm@usal.es'
fi
EOF

sudo chmod u+x /root/login-mail.sh

sudo tee -a /etc/pam.d/common-session > /dev/null <<'EOF'
session optional      pam_exec.so /bin/bash /root/login-mail.sh
EOF

sudo tee -a /etc/pam.d/common-session-noninteractive > /dev/null
<<'EOF'
session optional      pam_exec.so /bin/bash /root/login-mail.sh
EOF
```

### 2.14.2. Bloquear páginas web a los alumnos

Para bloquear Facebook y YouTube a los alumnos hemos configurado el servidor como un servidor proxy con Squid al que se deberían conectar los equipos de los alumnos:

```
sudo apt install squid
sudo systemctl restart squid
sudo systemctl enable squid
```

Luego editaremos la configuración para activar el bloqueo de páginas web:

```
sudo nano /etc/squid/squid.conf
acl block dstdomain "/etc/squid/website_block.txt"
http_access deny block
```

Permitimos las conexiones desde la misma LAN a Squid:

```
sudo sed -i 's/#http_access allow localnet/http_access allow
localnet/g' /etc/squid/squid.conf
```

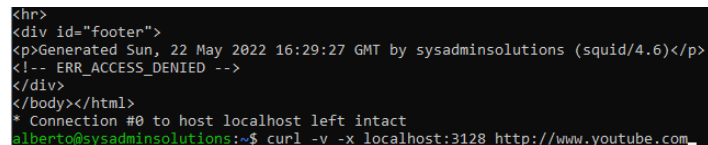
Y añadimos los dominios en /etc/squid/website\_block.txt

```
sudo tee /etc/squid/website_block.txt > /dev/null <<'EOF'
.facebook.com
.youtube.com
EOF
```

Reiniciamos Squid para aplicar los cambios:

```
sudo systemctl restart squid
```

Ahora los ordenadores de los alumnos se pueden configurar para que usen como proxy el servidor en su puerto 3128 y tendrán el acceso a YouTube y Facebook bloqueado:



```
<hr>
<div id="footer">
<p>Generated Sun, 22 May 2022 16:29:27 GMT by sysadminsolutions (squid/4.6)</p>
<!-- ERR_ACCESS_DENIED -->
</div>
</body></html>
* Connection #0 to host localhost left intact
alberto@sysadminsolutions:~$ curl -v -x localhost:3128 http://www.youtube.com_
```

### 2.14.3. Software de llamadas y videollamadas

Para que los usuarios puedan realizar llamadas y videollamadas instalaremos Jitsi Meet.

Primero añadiremos los repositorios necesarios:

```
echo deb http://packages.prosody.im/debian $(lsb_release -sc) main
| sudo tee -a /etc/apt/sources.list
wget https://prosody.im/files/prosody-debian-packages.key -O- |
sudo apt-key add -
curl https://download.jitsi.org/jitsi-key.gpg.key | sudo sh -c 'gpg
--dearmor > /usr/share/keyrings/jitsi-keyring.gpg'
echo 'deb [signed-by=/usr/share/keyrings/jitsi-keyring.gpg]
https://download.jitsi.org stable/' | sudo tee
/etc/apt/sources.list.d/jitsi-stable.list > /dev/null
sudo apt update
```

E instalamos el paquete:

```
sudo apt install jitsi-meet
```

La configuración de Jitsi en Nginx hace conflicto con la nuestra por lo que tenemos que quitar la primera línea del archivo `/etc/nginx/sites-enabled/meet.sysadminsolutions.cf.conf`.

En la configuración introduciremos el nombre del dominio de Jitsi “meet.sysadminsolutions.cf” y seleccionaremos generar un nuevo certificado. Para crear el certificado haremos:

```
sudo /usr/share/jitsi-meet/scripts/install-letsencrypt-cert.sh
```

Ahora limitaremos el acceso a los usuarios del sistema, para ello editaremos el bloque de Nginx y la configuración de LemonLDAP como en los casos anteriores:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
    'locationRules/meet.sysadminsolutions.cf' 'default'
'accept' \
    'locationRules/meet.sysadminsolutions.cf'
'(?#Logout)^\/logout/' 'logout_sso' \
    'exportedHeaders/meet.sysadminsolutions.cf' 'mai' '$mail' \
    'exportedHeaders/meet.sysadminsolutions.cf' 'displayName' '$cn'
```

```
location = /lmauth {
    internal;
    include /etc/nginx/fastcgi_params;
    fastcgi_pass unix:/var/run/llng-fastcgi-server/llng-
fastcgi.sock;
    fastcgi_pass_request_body off;
    fastcgi_param CONTENT_LENGTH "";
    fastcgi_param HOST $http_host;
    fastcgi_param X_ORIGINAL_URI $original_uri;
}

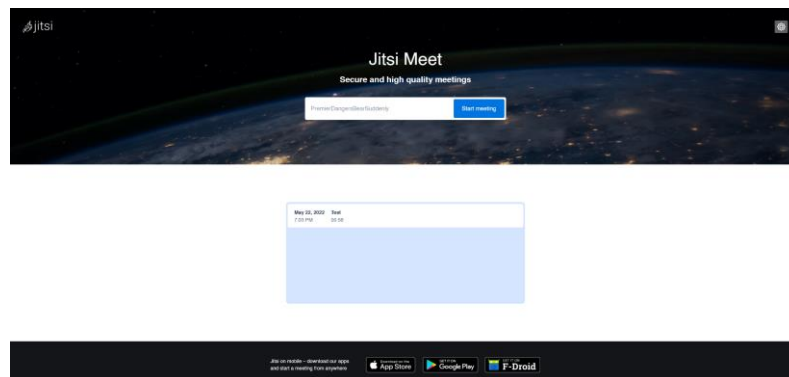
location / {
    auth_request /lmauth;
    set $original_uri $uri$is_args$args;
    auth_request_set $lmremote_user $upstream_http_lm_remote_user;
    auth_request_set $lmlocation $upstream_http_location;
    error_page 401 $lmlocation;

    auth_request_set $mail $upstream_http_mail;
    proxy_set_header mail $mail;
    auth_request_set $displayname $upstream_http_displayName;
    proxy_set_header displayName $displayname;
    auth_request_set $lmcookie $upstream_http_cookie;
    proxy_set_header Cookie: $lmcookie;
}
```

```
sudo nginx -t && sudo systemctl restart nginx
```

Añadimos la aplicación al portal:

```
sudo /usr/share/lemonldap-ng/bin/lemonldap-ng-cli -yes 1 \
  addKey \
    applicationList/1sample/meet type application \
    applicationList/1sample/meet/options description "Inicia
una llamada" \
    applicationList/1sample/meet/options display "on" \
    applicationList/1sample/meet/options logo "thumbnail.png" \
    applicationList/1sample/meet/options name "Videollamadas" \
    applicationList/1sample/meet/options uri
"https://meet.sysadminsolutions.cf/"
```



#### 2.14.4. Almacenar copias de seguridad en un servidor remoto

Para almacenar las copias de seguridad en un servidor remoto usaremos Rclone, que enviará el directorio con la copia de seguridad a MEGA. Instalamos Rclone:

```
curl https://rclone.org/install.sh | sudo bash
rclone config
```

En la configuración introduciremos nuestras credenciales de MEGA. Moveremos el fichero de configuración al home del usuario backuppc para que pueda llamar a rclone:

```
sudo mkdir -p /var/lib/backuppc/.config/rclone/
sudo mv ~/.config/rclone/rclone.conf
/var/lib/backuppc/.config/rclone/
sudo chown -R backuppc:backuppc /var/lib/backuppc/.config
```

Configuraremos BackupPC para que llame a rclone después de finalizar la copia de seguridad:

```
sudo tee -a /etc/backuppc/sysadminsolutions.cf.pl > /dev/null <<'EOF'
$Conf{DumpPostUserCmd} = 'rclone copy /var/lib/backuppc/cpool
mega:backups';
EOF
```

Ahora la copia de seguridad será replicada en MEGA cada vez que se realice.

## Bibliografía

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mariadb-php-lemp-stack-on-debian-10>

<https://wiki.debian.org/LDAP/OpenLDAPSetup>

<https://wiki.debian.org/LDAP/NSS>

<https://wiki.debian.org/LDAP/PAM>

<https://fusiondirectory-user-manual.readthedocs.io/en/latest/fusiondirectory/install/debian/debian-fd-install.html>

<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-debian-10>

<https://lemonldap-ng.org/documentation/latest/quickstart.html>

<https://github.com/Worteks/ldapcon2019-llng-workshop>

<https://www.linuxbabe.com/mail-server/build-email-server-from-scratch-debian-postfix-smtp>

<https://docs.sendgrid.com/for-developers/sending-email/postfix>

<https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lemp-nginx-mariadb-and-php-on-debian-10>

<https://wordpress.org/support/article/create-a-network/>

<https://oracle-patches.com/en/web/how-to-use-wordpress-multisite-with-nginx>

<https://sehnryr.medium.com/installing-filestash-on-debian-10-buster-8b6d33c8daed>

[https://en.wikibooks.org/wiki/OpenSSH/Cookbook/File\\_Transfer\\_with\\_SFTP#Three\\_Ways\\_of\\_Setting\\_Permissions\\_for\\_Chrooted\\_SFTP-Only\\_Accounts](https://en.wikibooks.org/wiki/OpenSSH/Cookbook/File_Transfer_with_SFTP#Three_Ways_of_Setting_Permissions_for_Chrooted_SFTP-Only_Accounts)

[https://docs.moodle.org/400/en/Installation\\_quick\\_guide](https://docs.moodle.org/400/en/Installation_quick_guide)

[https://www.zabbix.com/download?zabbix=6.0&os\\_distribution=debian&os\\_version=10\\_buster&db=mysql&ws=nginx](https://www.zabbix.com/download?zabbix=6.0&os_distribution=debian&os_version=10_buster&db=mysql&ws=nginx)

<https://mariadb.com/docs/service-management/upgrades/community-server/release-series-cs10-5-debian10/>

[https://www.zabbix.com/documentation/current/en/manual/appendix/install/web\\_service](https://www.zabbix.com/documentation/current/en/manual/appendix/install/web_service)

<https://backuppc.github.io/backuppc/BackupPC.html#Step-1:-Getting-BackupPC>

<https://www.digitalocean.com/community/tutorials/how-to-set-filesystem-quotas-on-debian-10>

[https://github.com/shartge/pam\\_setquota](https://github.com/shartge/pam_setquota)

<https://dev.to/dcodeyt/creating-beautiful-html-tables-with-css-428l>



<https://blog.stalkr.net/2010/11/login-notifications-pamexec-scripting.html>

<https://www.cloudbooklet.com/how-to-install-and-setup-sendmail-on-debian-10/>

<https://cloudinfrastructureservices.co.uk/how-to-block-websites-using-squid-proxy-server/>

<https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-quickstart/>

<https://rclone.org/install/>