PRÁCTICAS
DE
ADMINISTRACIÓN
DE SISTEMAS OPERATIVOS Y
REDES DE COMPUTADORES

HITO III

INSTALACIÓN DE SERVICIOS AVANZADOS SOBRE SERVIDORES:
Administración remota y monitorización de servicios

Alejandro Reyes Albillar
45931406-S
ara65@alu.ua.es
Curso 2015 / 2016
Grado en Ingeniería Informática

Universitat d'Alacant
Universidad de Alicante

ÍNDICE

1. Mensajería instantánea

Jabber, ahora conocido como XMPP, es un protocolo abierto y extensible basado en XML, originalmente ideado para mensajería instantánea.

Con el protocolo XMPP queda establecida una plataforma para el intercambio de datos XML que puede ser usada en aplicaciones de mensajería instantánea. Las características en cuanto a adaptabilidad y sencillez del XML son heredadas de este modo por el protocolo XMPP.

A diferencia de los protocolos propietarios de intercambio de mensajes como ICQ, Y! y Windows Live Messenger, se encuentra documentado y se insta a utilizarlo en cualquier proyecto. Existen servidores y clientes libres que pueden ser usados sin coste alguno.

Tras varios años de su existencia, ha sido adoptado por empresas como Facebook, Tuenti, WhatsApp Messenger y Nimbuzz, entre otras, para su servicio de chat. Google lo adoptó para su servicio de mensajería Google Talk, y en 2013 anunció que lo abandonaría en favor de su protocolo propietario Hangouts.

Para instalar este protocolo es necesario tener instalado java, ya que está basado en él. Además, haremos uso del programa openfire, el cual podemos obtener desde su página oficial para windows o mediante las órdenes "pkg install openfire" y "yum -y install openfire" para FreeBSD y CentOS respectivamente.

Este programa actúa como servidor para diferentes protocolos entre los que se encuentra XMPP, por lo que es uno de los más indicados para la instalación de este servicio.

Para la instalación en CentOS seguiremos los siguientes pasos:

- Asumiendo que la instalación de Centos 7 es "minimal", necesitaremos instalar wget para poder descargar el archivo rmp de Openfire, por tanto utilizaremos el comando "`yum install -y wget`".

- Creamos la base de datos con nombre "openfire" con PosgreSQL. Probablemente aparecerán algunas advertencias similares a "could not change directory to /root" los cuales ignoraremos:

```
createdb openfire
```

- Una vez creada la base de datos, creamos el usuario "openfire":

```
createuser -P openfire
```

- Inmediatamente después nos pedirá ingresar dos veces una contraseña para este usuario “Enter password for new role:”. Ingresa una contraseña fuerte. Este es un buen sitio para generar contraseñas seguras: https://strongpasswordgenerator.com/
- Ahora es necesario setear el usuario administrador de PostgreSQL:

```
psql -U postgres -d postgres -c "ALTER USER postgres WITH
PASSWORD 'CONTRASEÑA-INGRESADA-EN-EL-PASO-ANTERIOR';"
```

- Las contraseñas están seteadas, ahora configuraremos la base de datos de modo tal que para cada conexión se requiera el ingreso del password

```
vi /var/lib/pgsql/data/pg_hba.conf
```

- Ir al final del archivo y modificar los datos y modificar “ident” y “peer” por “md5”, luego guardar el archivo.

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only

local   all         all                               md5

# IPv4 local connections:

host    all         all         127.0.0.1/32          md5

# IPv6 local connections:

host    all         all         ::1/128               md5
```

- Salir de la línea de comando de PostgreSQL con “`exit`” y reiniciamos la base de datos con “`systemctl restart postgresql.service`“.

Instalación de Openfire

- Primero descargamos el .rpm de Openfire en /root:

```
wget
http://www.igniterealtime.org/downloadServlet?filename=openfire/
openfire-3.10.2-1.i386.rpm /root/openfire-3.10.2-1.i386.rpm
```

- Ahora instalamos el paquete:

```
yum install -y /root/downloadServlet?filename=openfire/openfire-
3.10.2-1.i386.rpm
```

- Este paquete de instalación contiene su proio JRE por lo tanto no será necesario instalarlo adicionalmente. Lo que si necesitamos instalar son algunas librerias de x84 ya que la versión de Openfire solo está disponible en 32 bits (asumiendo que nuestro Centos 7 es x64)

```
yum install -y glibc.i686
```

- Ahora seteamos el inicio automático de Openfire después de cada reinicio:

```
chkconfig openfire on
```

- E iniciamos el servicio:

```
systemctl start openfire.service
```

- Agregamos dos reglas el firewall para permitir las conexiones a Openfire

```
firewall-cmd --permanent --zone=public --add-port=9090/tcp
firewall-cmd --permanent --zone=public --add-port=9091/tcp
firewall-cmd --permanent --zone=public --add-port=5222/tcp
firewall-cmd --permanent --zone=public --add-port=5223/tcp
```

- Reiniciamos el servicio de firewall con "`firewall-cmd --reload`".

- Ahora que ya instalamos el servidor, podemos proceder a configurarlo mediante la interfaz gráfica. Para esto debes abrir un browser desde un computador con acceso a la red donde se instaló el servidor, accediendo a la dirección http://ip-servidor:9090. Normalmente accederemos desde el mismo servidor con localhost:9090.

- Primero debemos elegir un idioma. En inglés es más fácil encontrar documentación en caso de necesitarla pero podemos escoger el idioma que deseemos.

- A continuación seteamos el dominio de nuestra instancia. Por favor reemplazar OUR-CHOOSEN-DOMAIN.com por un nombre de dominio más apropiado.

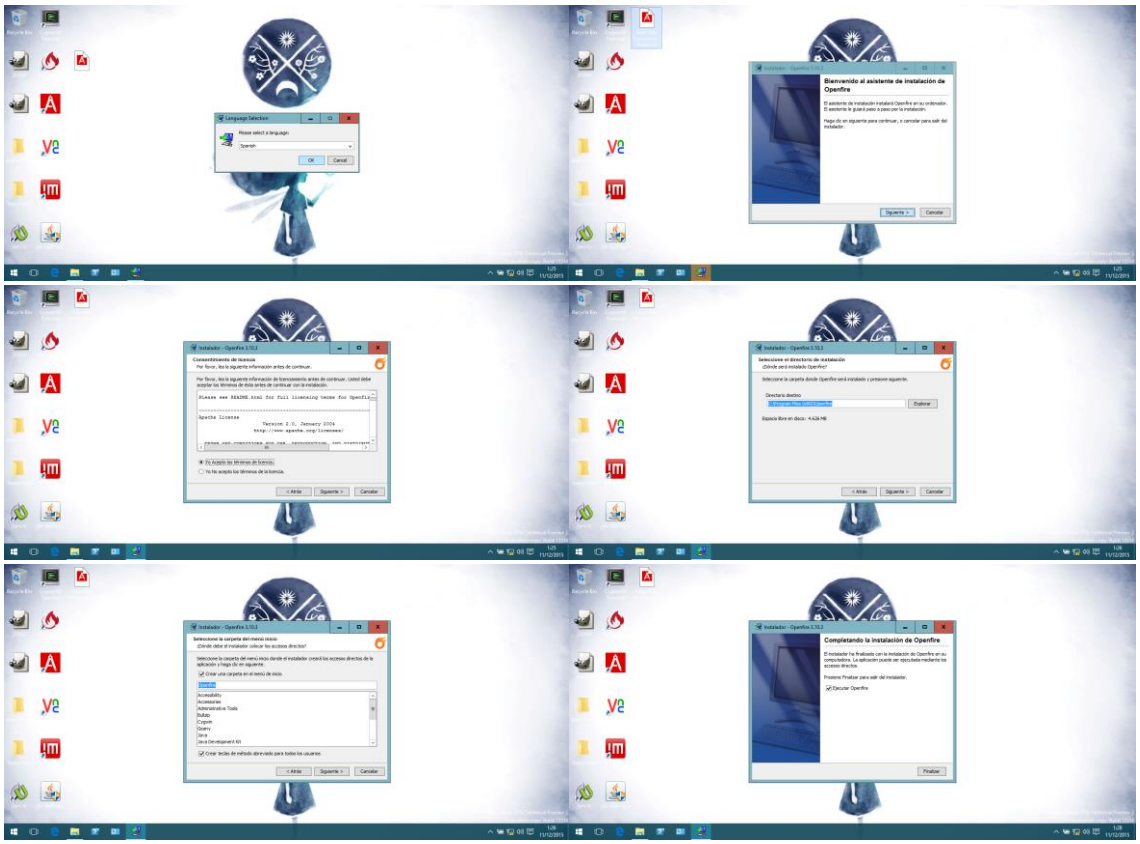
- En el siguiente paso, tras seleccionar conexión standard, debemos elegir "PostgreSQL" como motor de base de datos, ingresar usuario y contraseña creados anteriormente.
  Donde dice JDBC Driver Class ponemos la dirección "org.postgresql.Driver" y donde pone database URL ponemos la dirección de la base de datos creada, que es "jdbc:postgresql://localhost:5432/openfire".

- En el siguiente paso debemos elegir el método de autentificación de usuarios. Este paso es de libre elección pero nosotros dejaremos la autentificación por defecto.

- En el paso final debemos ingresar un nombre de usuario y contraseña para el administrador de la interfaz web de nuestra instancia Openfire.

- Finalmente podemos ingresar al administrador web ingresando las credenciales elegidas en el paso anterior.

Para la instalación en Windows ejecutaremos el instalador que descargamos de la página oficial de Openfire:



La versión para Windows no incluye Java, por lo que deberemos instalar anteriormente el último JDK disponible descargable desde la página de Oracle.

Una vez instalado Openfire, abrimos la página de configuración localhost:9090 y configuramos el servidor como lo hemos hecho con CentOS. Sin embargo la base que utilizaremos en Windows es una base de datos de Oracle por lo que la dirección de la base de datos deberemos adecuarla al nombre de base de datos y credenciales que hemos creado anteriormente. Es por esto, por lo que deberemos añadir, al directorio /lib, los drivers de Oracle express (ojdb6.jar) descargables descargables desde su página.

Deberemos crear un nuevo usuario y para eso debemos crear un nuevo TableSpace:

```
CREATE TABLESPACE MyTableSpace DATAFILE
'C:\oraclexe\app\oracle\oradata\XE\MyTableSpace.DBF' SIZE 30M;

CREATE USER ara65 IDENTIFIED BY ara1995 DEFAULT TABLESPACE
MyTableSpace QUOTA 10M ON MyTableSpace;

GRANT dba, connect, resource TO ara65;

GRANT connect, create session TO ara65;
```

Una vez creado tanto el TableSpace como el usuario, utilizamos la base proporcionada por el programa, situada en el directorio C:\Program Files (x86)\Openfire\resources\database\openfire_oracle.sql para crear las tablas que necesita openfire para funcionar.

Ahora solamente hay que seguir la configuración con el nuevo usuario y listo.
Finalmente, para instalar el servicio en FreeBSD habremos creado antes, en MySQL, una base de datos y el servicio de OpenFire con el comando "`pkg install openfire`".
Debemos iniciar el servicio con "service openfire start" pero antes debemos añadir a "/etc/rc.conf" la línea:

- `openfire_enable="YES"`

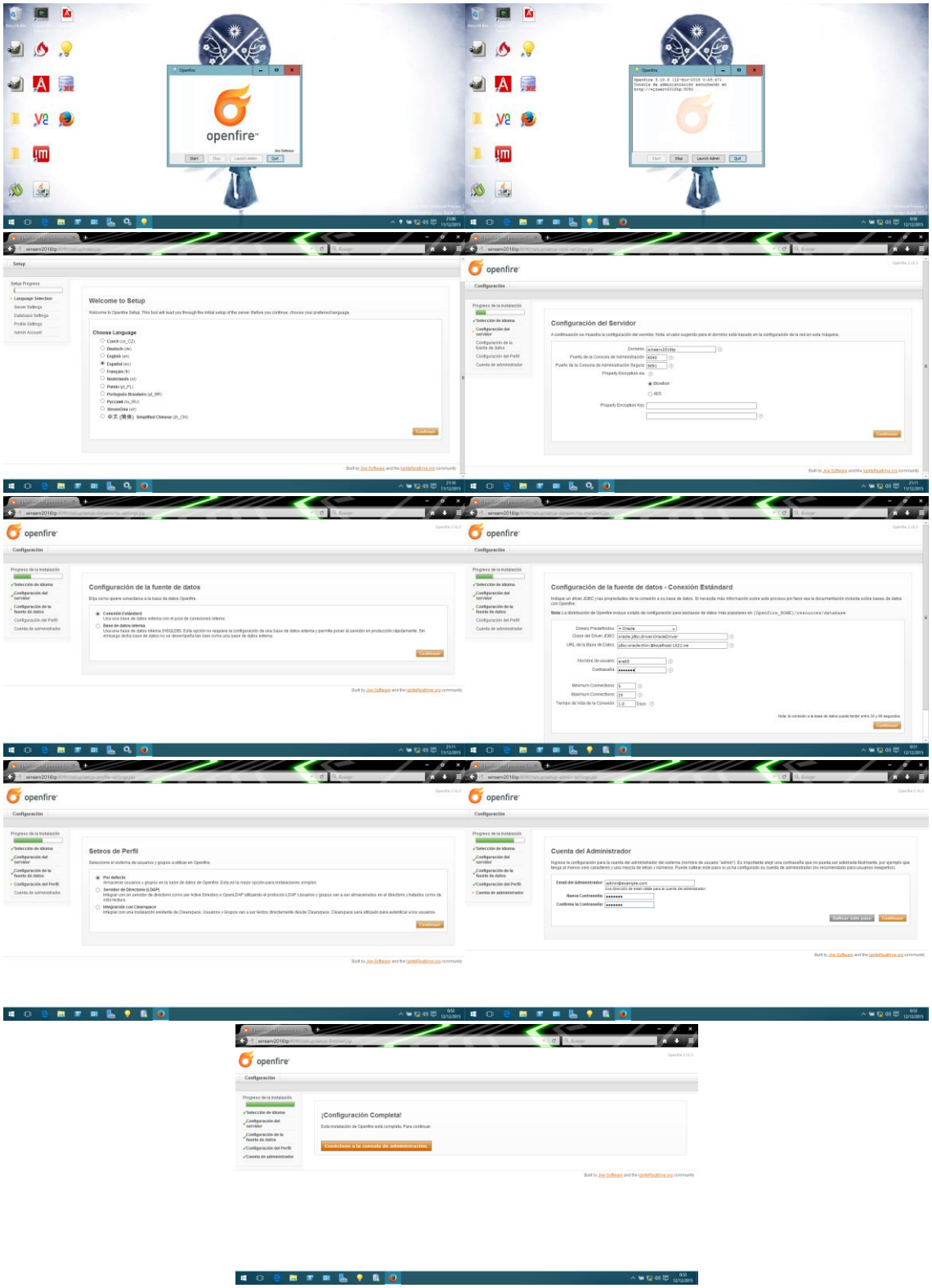
Crearemos ahora la base de datos de MySQL:

- `create database openfire;`
- `grant all privileges on openfire.* to openfire@localhost identified by 'ara1995';`

Creamos el archivo openfire_mysql.sql en el directorio "/usr/local/etc/openfire" y lo rellenamos con el código que podemos encontrar en esta página. Este archivo tiene el constructor de todas las tablas necesarias para que funcione openfire.

Ahora ejecutamos el script sql sobre la base de datos con "cat openfire_mysql.sql | mysql openfire". En el caso de que aparezca un error, es debido a que la tabla "ofRoster" tiene un elemento que ha sido definido como "VARCHAR(1024)" cuando dicha clase no admite más de 767 bytes de tamaño. Debemos asegurarnos de cambiar este valor antes de ejecutar la orden anterior para evitar errores.

Finalmente configuramos mediante el asistente web como hemos realizado anteriormente, escogiendo esta vez MySql e introduciendo los parámetros correspondientes para que funcione correctamente.



Existen múltiples clientes jabber que podemos utilizar para conectarnos a nuestro servidor, aunque nosotros utilizaremos Spark.

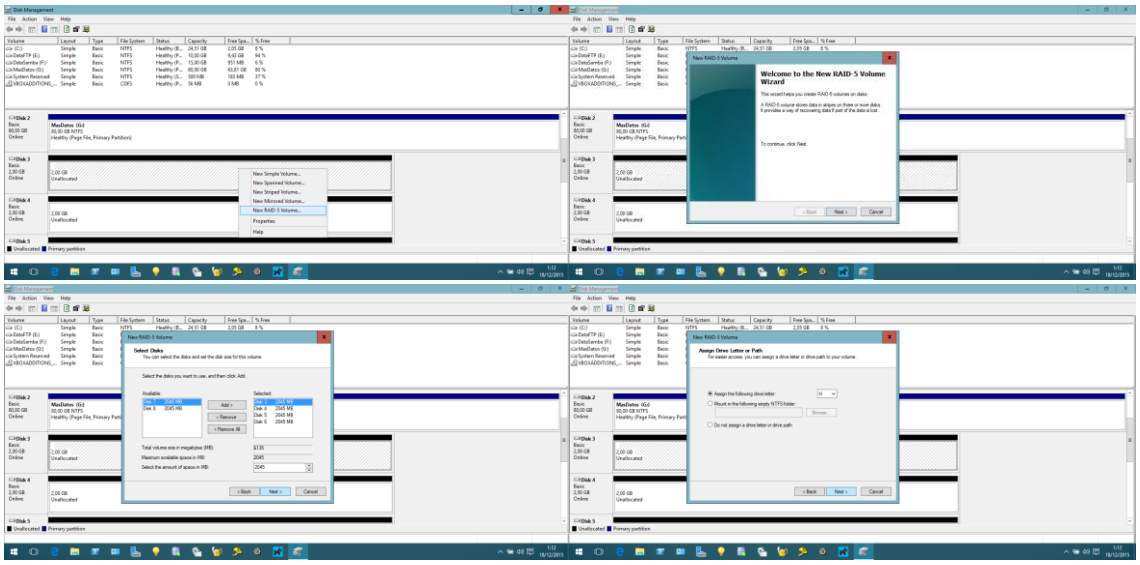
2. RAID

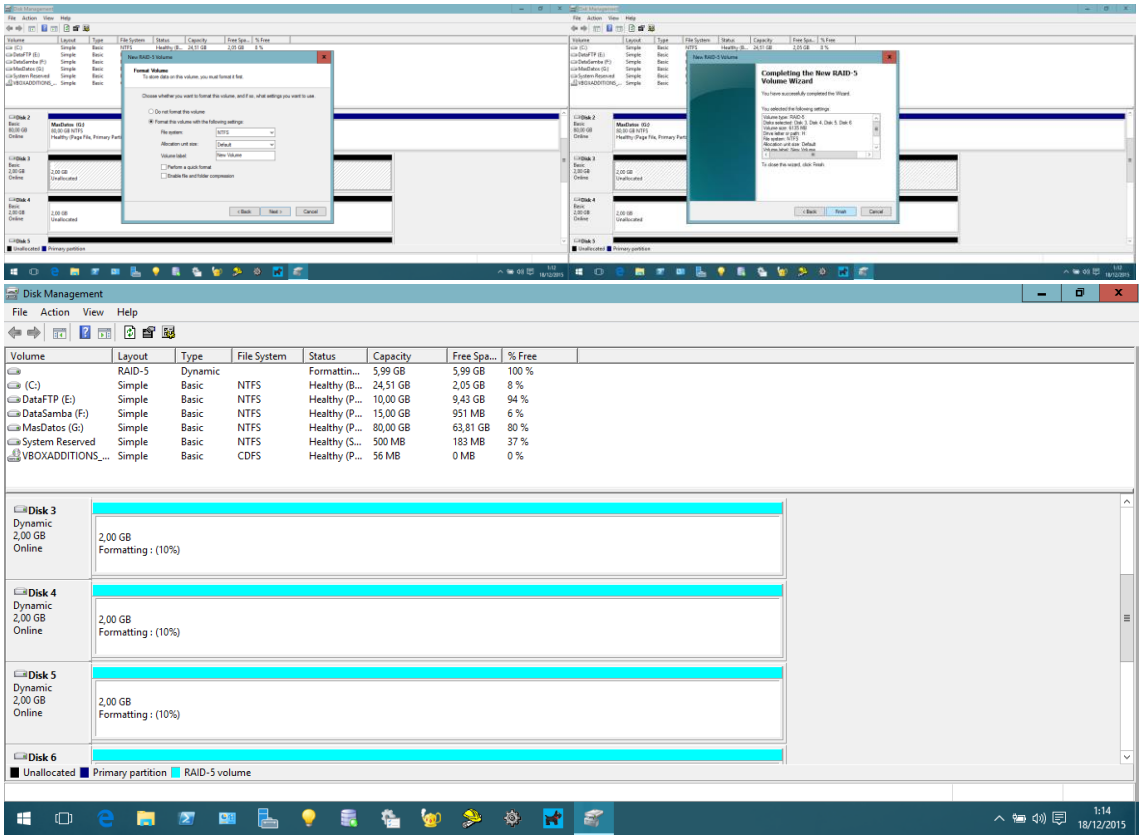
Para instalar RAID, ya sea de tipo 1, 5 o 10, que son los solicitados en esta práctica utilizaremos, en CentOS, el programa “mdadm”, el cual nos permite crear las particiones de manera automática indicando únicamente el nivel con el que se desea realizar la configuración.

Es por esto que para la configuración en CentOS utilizamos la orden “mdadm --create /dev/nombreRaid --level=X --raid-devices=Y </dev/disk1, /dev/disk2, ···, /dev/diskY>”, donde “X” es el nivel del raid a crear e “Y” el número de discos añadidos al principio al raid. Más adelante se pueden añadir nuevos discos con “mdadm -a /dev/nombreRaid /dev/newDisk” o quitarlos cambiando la opción “-a” por “-r”. De igual modo, se pueden marcar como faltantes los discos mediante la opción “-f”. Esta opción marca el disco utilizado como faltante, es decir, lo extrae del raid sin quitarlo de su configuración, de modo que podamos insertar uno nuevo que no lleve ficha marca, mientras que “-r” elimina el disco de la configuración del raid, lo que hace que, en el caso de que se usarán 3 discos, ahora sólo se usen 2 y no se necesite de un tercero. Para la configuración en Windows utilizaremos el Administrador de discos que nos proporciona la interfaz del sistema. Dicha interfaz nos permite crear tanto RAID 1 (Mirror) como RAID 5. Sin embargo, Windows Server 2016 Technical Preview no tiene implementada la opción para crear RAID 10 por software como la tenían sus predecesores, por decisión de la empresa. De modo que para realizar un RAID 10 se debe realizar por Hardware de manera obligatoriamente.

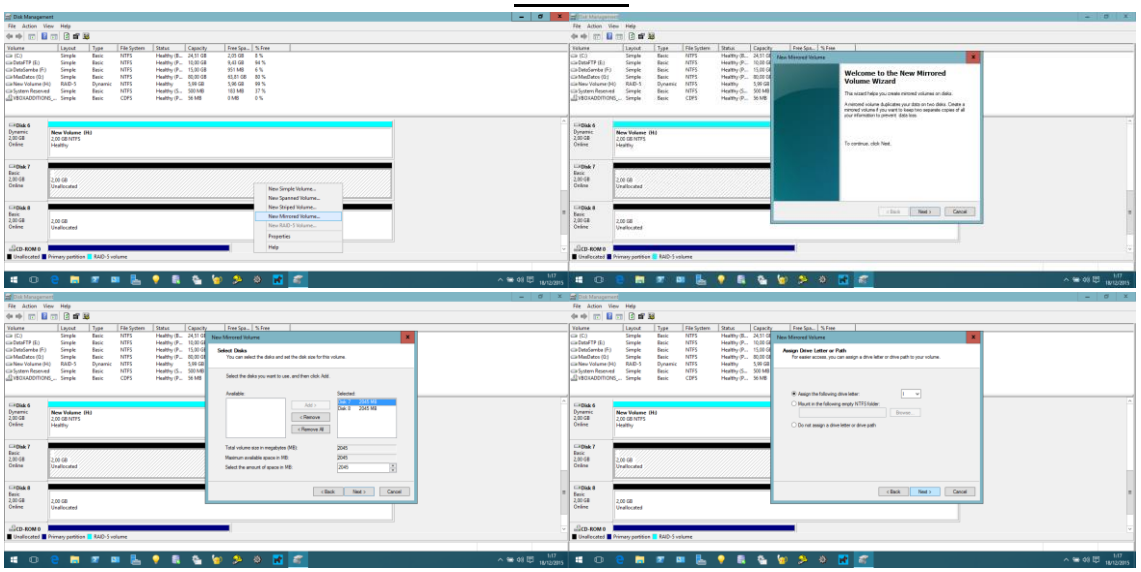
Los pasos a seguir para montar un RAID son básicamente los mismos, únicamente cambia la cantidad de discos a añadir, ya que depende del usuario, y el menú que selecciona el tipo de RAID a montar.

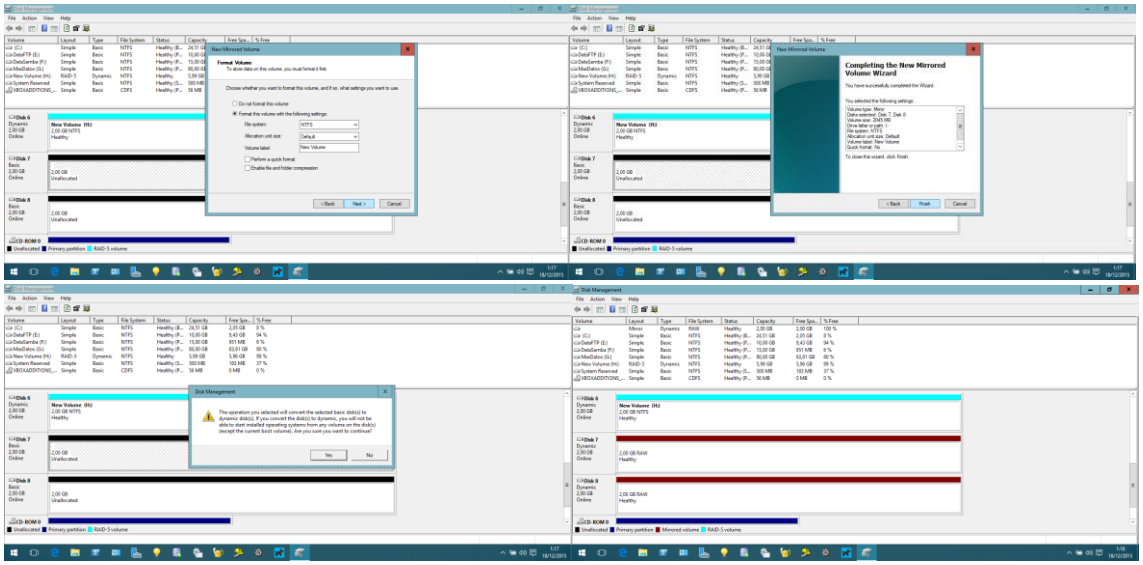
RAID 5




Disk Management
File Action View Help
Volume Layout Type File System Status Capacity Free Spa... % Free
RAID-5 Dynamic Formattin... 5,99 GB 5,99 GB 100 %
(C:) Simple Basic NTFS Healthy (B... 24,51 GB 2,05 GB 8 %
DataFTP (E:) Simple Basic NTFS Healthy (P... 10,00 GB 9,43 GB 94 %
DataSamba (F:) Simple Basic NTFS Healthy (P... 15,00 GB 951 MB 6 %
MasDatos (G:) Simple Basic NTFS Healthy (P... 80,00 GB 63,81 GB 80 %
System Reserved Simple Basic NTFS Healthy (S... 500 MB 183 MB 37 %
VBOXADDITIONS_... Simple Basic CDFS Healthy (P... 56 MB 0 MB 0 %
Disk 3
Dynamic
2,00 GB
Online
2,00 GB
Formatting : (10%)
Disk 4
Dynamic
2,00 GB
Online
2,00 GB
Formatting : (10%)
Disk 5
Dynamic
2,00 GB
Online
2,00 GB
Formatting : (10%)
Disk 6
Unallocated Primary partition RAID-5 volume
1:14
18/12/2015

RAID 1





3. Servidor Web

Tanto en CentOS como en FreeBSD vamos a instalar Apache Web Server como servidor web. Para hacer esto en CentOS utilizaremos el comando “yum -y install httpd”.

Deberemos seguir los siguientes pasos para configurar correctamente nuestro servidor para que funcione con php, html y dominios virtuales:
Borramos la página de bienvenida creada por defecto con el comando “`rm -f /etc/httpd/conf.d/welcome.conf`”.

Configuramos el archivo “`/etc/httpd/conf/httpd.conf`” modificando lo siguiente:

- Cambiamos la línea del email del administrador:
  - `ServerAdmin ara65@ara65.centos.com`
- Cambiamos el nombre del servidor.
  - `ServerName ara65.centos.com:80`
- Cambiamos en la línea 151 “AlowOverride” a “All”.
- Cambiar o añadir el nombre de los archivos a los que se pueden acceder mediante el nomre de directorio:
  - `DirectoryIndex index.html index.cgi index.php`
- Añadir las siguientes líneas al final:
  - `ServerTokens Prod`
  - `KeepAlive On`

Una vez hecho esto solamente debemos crear el archivo “index.html” en “/var/www/html” y comprobar que funciona.
Debemos recordar de abrir el puerto 80 para que se pueda acceder desde fuera al servicio:

- `firewall-cmd --permanent --zone=public --add-port=80/tcp`
- `firewall-cmd --reload`

Para instalar el servidor Web en FreeBSD utilizaremos el servidor Apache, como en CentOS, y lo instalaremos con el comando "`pkg install apache24`".

A continuación añadimos al archivo "/etc/rc.conf" la línea "`apache24_enable="YES"`" e iniciamos el servicio con el comando "`service apache24 start`".

Para instalar PHP utilizamos el comando "pkg install php56 mod_php56 php56-mysql".
Una vez instalado, debemos añadir al archivo /etc/local/apache24/httpd.conf las siguientes líneas:

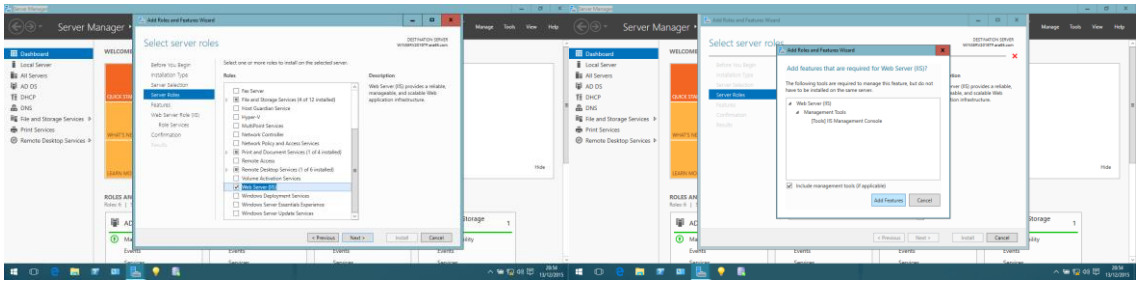
- `AddType application/x-httpd-php .php`
- `AddType application/x-httpd-php-source .phps`
- `LoadModule php5_module libexec/apache24/libphp5.so`

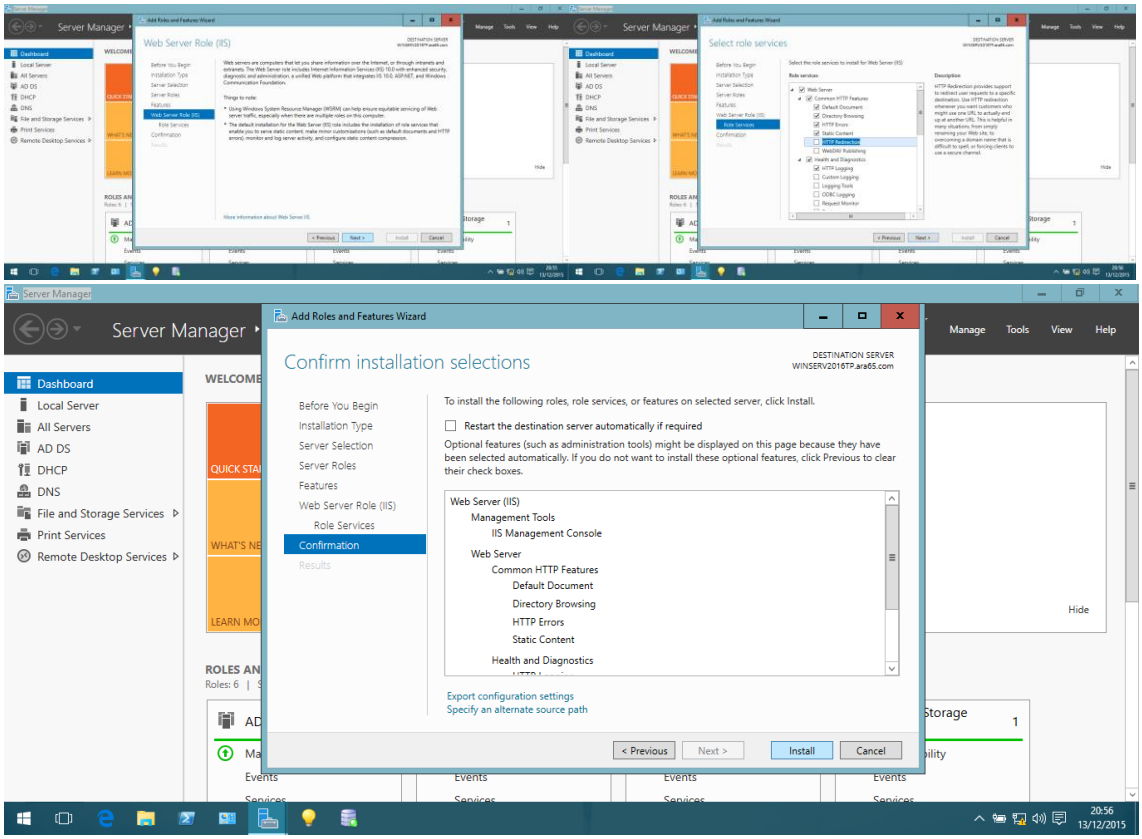
Y tenemos que asegurarnos de tener puesto "index.php" en la línea "DirectoryIndex". Las páginas que sirvamos se localizarán en el directorio "/usr/local/www/apache24/data".

Para instalar el host virtual debemos descomentar la línea "`Include etc/apache24/extra/httpd-vhosts.conf`" del archivo "`/usr/local/etc/apache24/httpd.conf`".

Además deberemos modificar el archivo "/etc/apache24/extra/httpd-vhosts.conf" y añadir el nombre del nuevo host a usar.

Por último, instalaremos el servidor en Windows mediante IIS. Para ello, antes debemos añadir el rol en el service manager.





Una vez instalado el rol, abrimos Administrative Tools > Internet Information Server (IIS) Manager y seleccionamos nuestro servidor.
Crearemos un archivo HTML y lo introduciremos en la carpeta “C:\inetpub\wwwroot”.

Añadimos PHP desde el programa Web PI de Microsoft, el cual podemos descargar desde [la página oficial de Microsoft](#).

Una vez hecho esto ya podemos acceder a las páginas PHP que introduzcamos en el directorio “C:\inetpub\wwwroot”.

4. Backup

Para instalar RSYNC en FreeBSD utilizaremos el comando “pkg install rsync” y añadimos la línea “`rsync_enable=”YES”`” al archivo “/etc/rc.conf”. A continuación editamos el fichero “/usr/local/etc/rsync/rsyncd.conf” para configurar nuestro servidor.

```
uid             = rsync
gid             = rsync
use chroot      = no
max connections = 4
syslog facility = local5
```

```
pid file        = /var/run/rsyncd.pid


[www]
        path    = /usr/local/websites/
        comment = all of the websites

auth users = ara65, root
secrets file = /usr/local/etc/rsync/rsyncd.secrets
```

Añadimos la siguiente línea al archivo “/etc/passwd”:

- `rsync:*:4002:4002::0:0:rsync daemon:/nonexistent:/sbin/nologin`

Y la siguiente a “/etc/group”:

- `rsync:*:4002:`

Creamos el archivo “/usr/local/etc/rsync/rsyncd.secrets” e introducimos “user:pass” en cada línea:

```
ara65:mypass
root:mypass
```

Una vez hecho esto iniciamos el servicio con “service rsyncd start” y comprobaamos con “telnet localhost 873” que se ha iniciado el servicio. Hecho esto ya está configurado el servidor RSYNC.

Para la instalación en CentOS, ejecutamos el comando “yum -y install rsync” y, una vez instalado, editamos el fichero “/etc/rsyncd.conf” del mismo modo que en FreeBSD para aceptar conexiones externas.

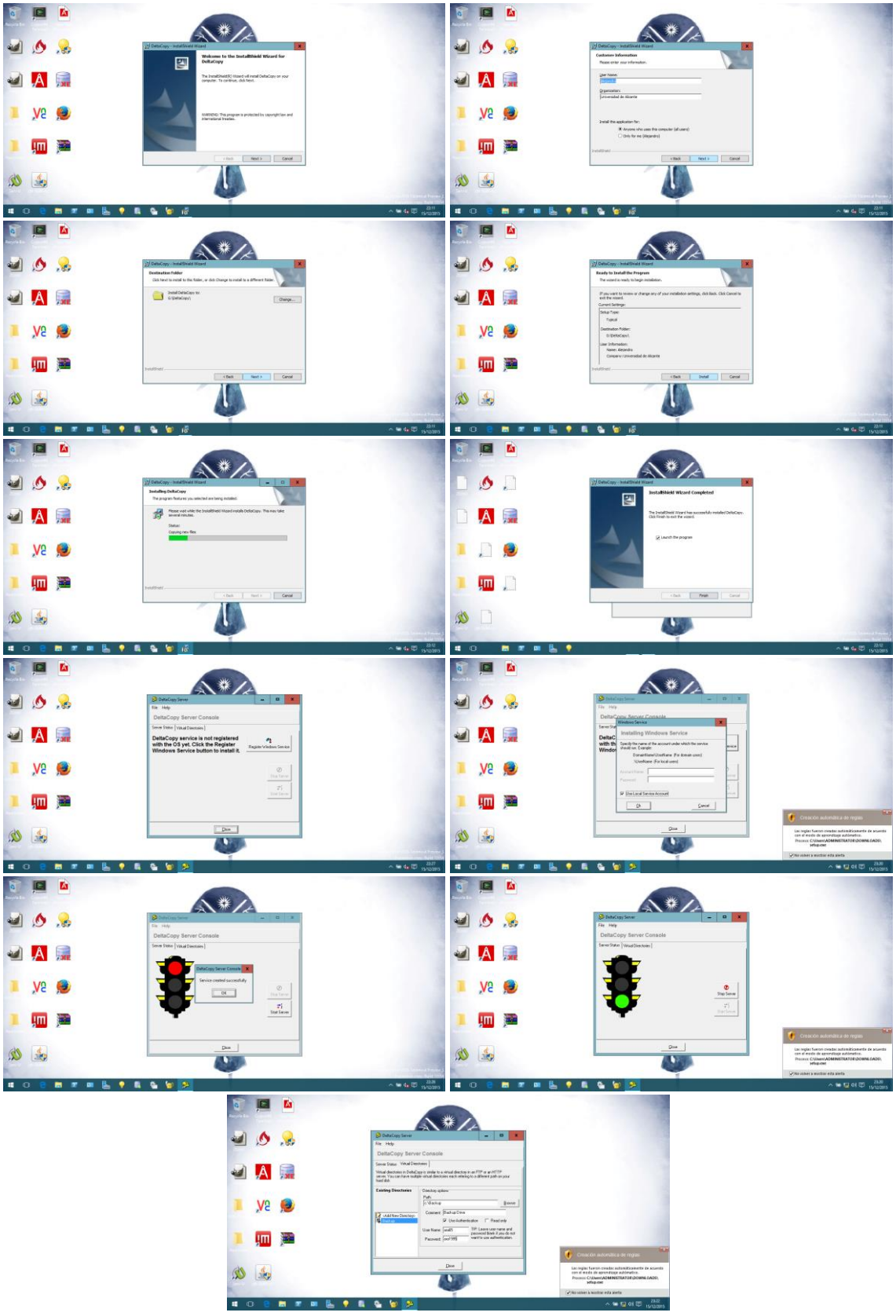
En CentOS el archivo “rsyncd.secrets” se colocará en la ruta “/etc/rsyncd.secrets”.

En caso de querer excluir archivos de la copia, indicaremos los nombres de estos archivos en el fichero “`/etc/rsync_exclude.lst`”.

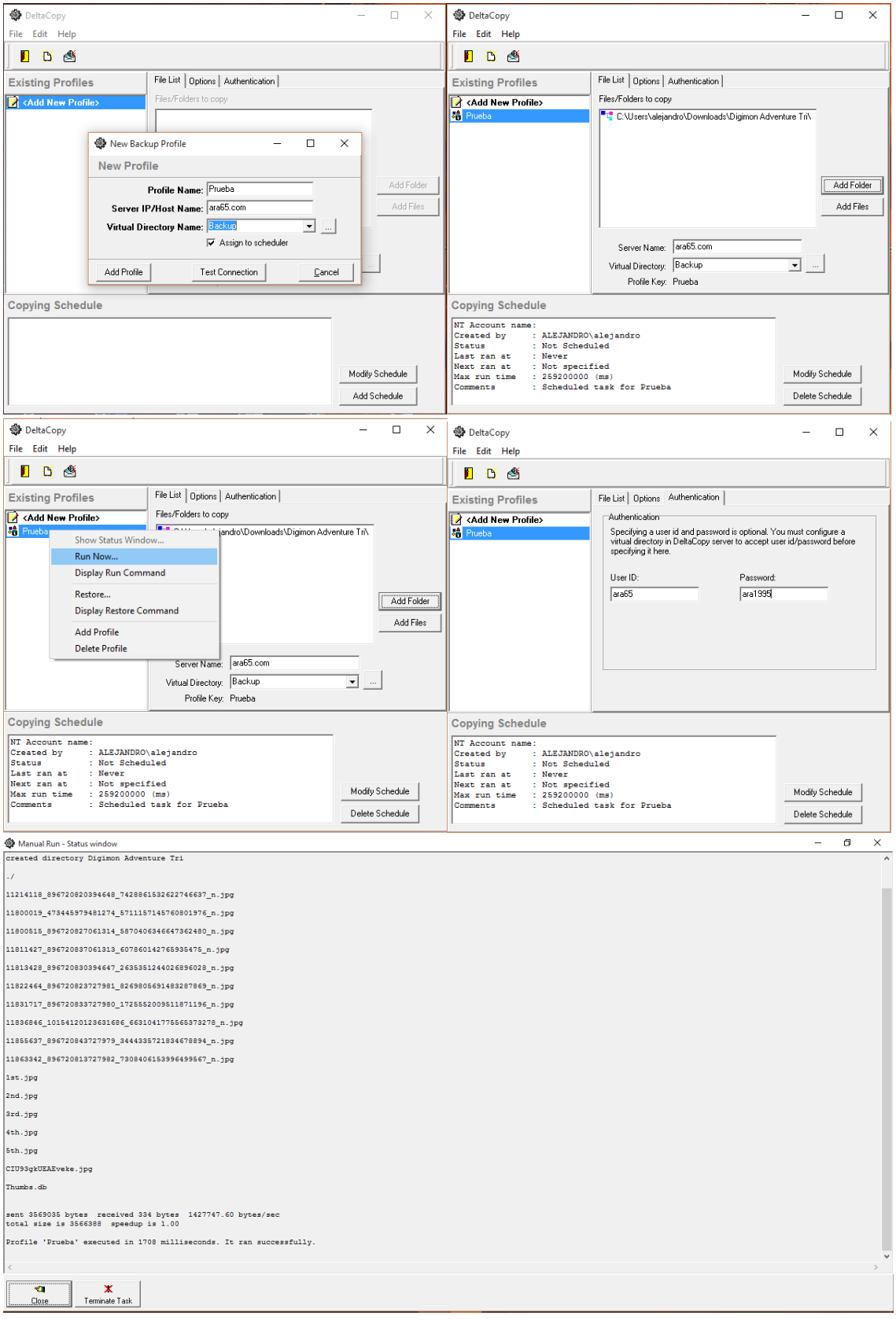
Finalmente, iniciamos el servicio con “systemctl start rsyncd” y lo habilitamos para que se inicie con el sistema con “systemctl enable rsyncd”.

En Windows instalaremos DeltaCopy, la cual podemos descargar desde su página.
En la página podemos ver una version sin instalador, que es el cliente, y otra con instalador, que es el servidor.

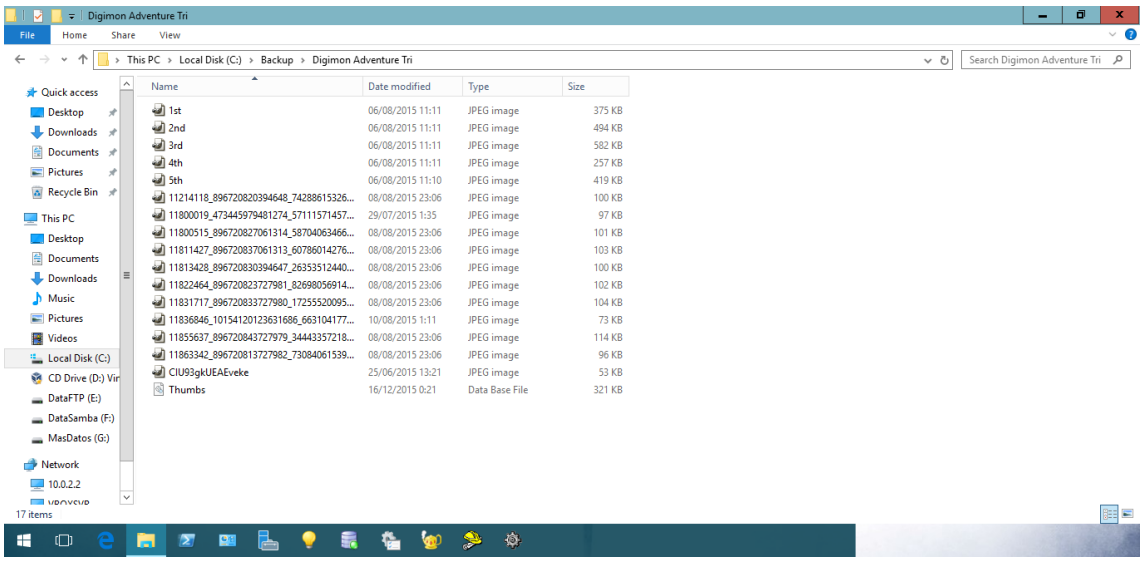
Una vez descargada, la instalamos, creamos el directorio en el que vamos a guardar la copia de seguridad y lo seleccionamos.


Welcome to the InstallShield Wizard for DeltaCopy
DeltaCopy Server Console
DeltaCopy service is not registered with the OS yet. Click the Register Windows Service button to install it.
Register Windows Service
Installing Windows Service
Service created successfully.
OK
Close

Nos conectamos desde el cliente con las credenciales proporcionadas.



Podemos observar que los archivos se han transmitido correctamente.



5. Servidor de BD: Mysql, PostgreSQL y Oracle express

Para instalar PostgreSQL en CentOS seguiremos los siguientes pasos:

- Instalaremos PostgreSQL para utilizar como nuestro motor de base de datos con el comando "`yum install -y postgresql postgresql-server postgresql-devel postgresql-libs`".

- Despues de completar la instalación nos aseguraremos que PostgreSQL inicie automáticamente después de los reinicios con el comando "`systemctl enable postgresql.service`".

- Luego inciamos la estructura de directorios y base de datos con "`postgresql-setup initdb`" e iniciamos el servicio con "`systemctl start postgresql.service`".

- Una vez instalado PostgreSQL, necesitamos crear la base de datos, usuario y contraseña. Para eso tenemos que cambiar de usuario con "`su postgres`".

- Creamos la base de datos con nombre "asorc". Probablemente aparecerán algunas advertencias similares a "could not change directory to /root" los cuales ignoraremos:

```
createdb asorc
```

- Una vez creada la base de datos, creamos el usuario "userasorc":

```
createuser -P userasorc
```

- Inmediatamente después nos pedirá ingresar dos veces una contraseña para este usuario “Enter password for new role:”. Ingresa una contraseña fuerte. Este es un buen sitio para generar contraseñas seguras: https://strongpasswordgenerator.com/

- Ahora es necesario setear el usuario administrador de PostgreSQL:

```
psql -U postgres -d postgres -c "ALTER USER postgres WITH
PASSWORD 'CONTRASEÑA-INGRESADA-EN-EL-PASO-ANTERIOR';"
```

- Las contraseñas están seteadas, ahora configuraremos la base de datos de modo tal que para cada conexión se requiera el ingreso del password

```
vi /var/lib/pgsql/data/pg_hba.conf
```

- Ir al final del archivo y modificar los datos y modificar “ident” y “peer” por “md5”, luego guardar el archivo.

```
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only

local   all         all                               md5

# IPv4 local connections:

host    all         all         127.0.0.1/32          md5

# IPv6 local connections:

host    all         all         ::1/128               md5
```

- Salir de la línea de comando de PostgreSQL con “`exit`” y reiniciamos la base de datos con “`systemctl restart postgresql.service`“.

- Ahora la base de datos ya está creada y podremos acceder a ella mediante HTTP.

- Finalmente ejecutamos el comando “`setsebool httpd_can_network_connect_db on`” para permitir a httpd conectar con las bases de datos

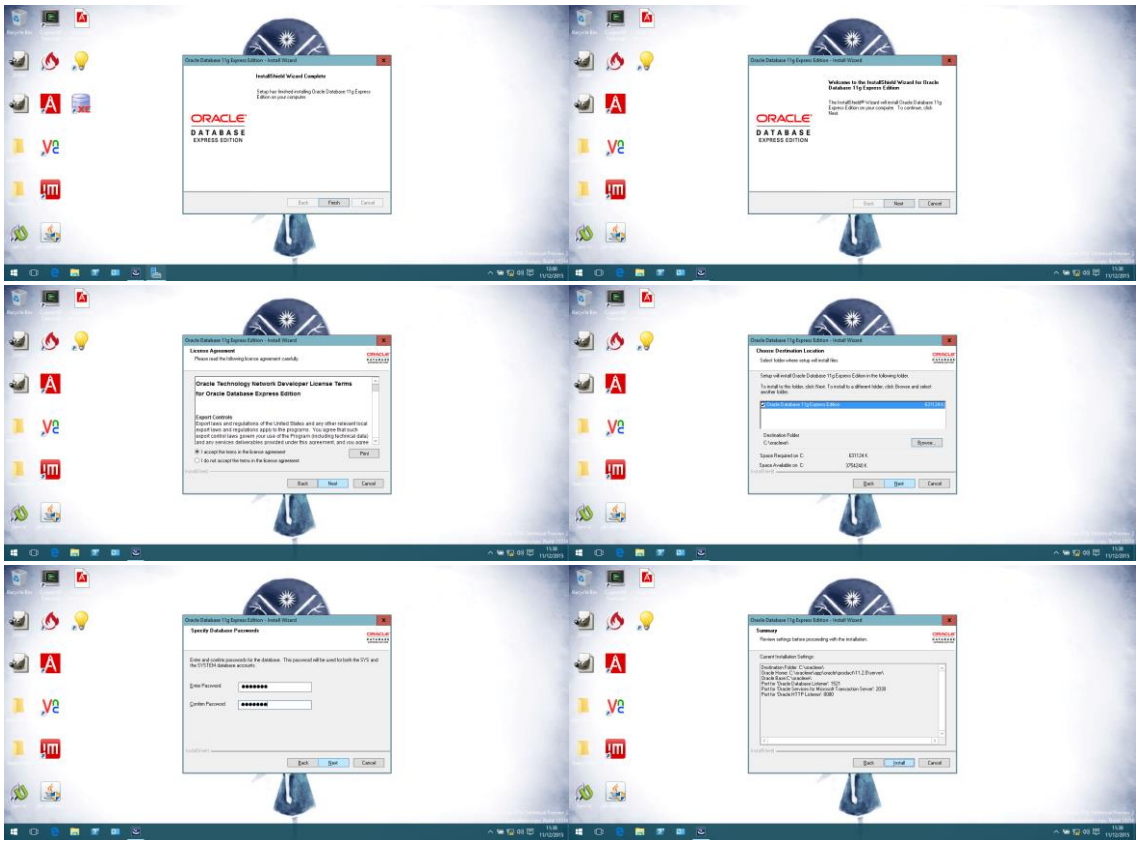
Para instalar PHP y poder utilizar los paquetes necesarios para utilizar postgreSQL instalaremos el paquete necesario con “yum -y install phpPgAdmin php-pgsql”.

Para instalar algunos de estos paquetes es necesario tener activado el repositorio epel así que, en el caso de no tenerlo instalado, lo instalaremos con las siguientes ordenes:

- `wget http://mirror.pnl.gov/epel/7/x86_64/e/epel-release-7-5.noarch.rpm`
- `rpm -Uvh epel-release-7-5.noarch.rpm`
- `yum repolist`
- `yum update`

Ahora que está instalado epel, procedemos con la instalación:

En Windows instalaremos Oracle 11g Express Edition como motor de base de datos. La descarga nos da un .zip que contiene una carpeta en la que se encuentra el instalador.



Para poder acceder mediante un usuario distinto a SYS, ya que no se permite su uso en muchos comandos, debemos crear un nuevo namespace con oracle y luego crear un usuario asignado a ese namespace.

```
CREATE TABLESPACE MyTableSpace DATAFILE
'C:\oraclexe\app\oracle\oradata\XE\MyTableSpace.DBF' SIZE 30M;
```

```
CREATE USER ara65 IDENTIFIED BY ara1995 DEFAULT TABLESPACE
MyTableSpace QUOTA 10M ON MyTableSpace;

GRANT dba, connect, resource TO ara65;

GRANT connect, create session TO ara65;
```

Una vez creados tanto el usuario como el namespace, nos autenticamos en la base de datos con las credenciales del nuevo usuario y creamos las tablas que veamos convenientes e insertamos los datos en ellas.

```
CREATE TABLE usuarios(id int, nombre varchar(600), correo
varchar(600));

INSERT INTO usuarios VALUES(1,'Alejandro','ara65@alu.ua.es');
```

En FreeBSD instalaremos MySQL server con el comando "pkg install mysql56-server". Tras la instalación añadiremos al archivo "/etc/rc.conf" la línea:

- `mysql_enable="YES"`

Finalmente iniciamos el servicio con "service mysql-server start".

Una vez instalado el servidor accedemos a la consola de mysql-server con el usuario root. Para ello escribimos "mysql -u root -p" y se nos pide que introduzcamos una contraseña, a lo cual simplemente le daremos a enter, al no haber establecido una contraseña para el administador.

Ahora crearemos una nueva tabla y un usuario que pueda acceder a dicha tabla con los siguientes comandos:

- `create database asorc;`
- `grant all privileges on asorc.* to ara65@localhost identified by 'ara1995';`

6. Proxy Cache

Para instalar el proxy en los 3 sistemas hemos elegido instalar Squid, el cual está basado en Unix/Linux pero también podemos descargarnos los binarios para Windows desde su página.

Para la instalación de Squid en CentOS utilizaremos la orden "yum -y install squid". Una vez instalado lo configuraremos editando el archivo "/etc/squid/squid.conf" de la siguiente manera:

- `acl CONNECT method CONNECT`

```
-   #Añadir un nuevo ACL (línea 26)
-   acl lan src 192.168.122.0/24
-   # Añadir los siguientes para Autenticación Básica
-   auth_param basic program /usr/lib64/squid/basic_ncsa_auth
    /etc/squid/.htpasswd
-   auth_param basic children 5
-   auth_param basic realm Squid Basic Authentication
-   auth_param basic credentialsttl 5 hours
-   acl password proxy_auth REQUIRED
-   http_access allow password

-   http_access allow localhost

-   # Permitir el nuevo ACL y a todas las conexiones http (línea 54)
-   http_access allow lan
-   http_access allow all

-   # Rechazar el resto de conexiones http entrantes
-   http_access deny all

-   # Cambiar puerto (línea 59)
-   http_port 8080

-   # Especificar backend del servidor web (línea 63)
-   http_port 80 accel defaultsite=www.ara65.centos.com

-   # Descomentar línea 72
-   # Significado de los números ⇒ [Tamaño de caché del disco] [nº de
    directorios en el nivel superior] [nº de directorios en el segundo
    nivel]
-   cache_dir ufs /var/spool/squid 100 16 256

-   # Añadir al final
-   request_header_access Referer deny all
-   request_header_access X-Forwarded-For deny all
-   request_header_access Via deny all
-   request_header_access Cache-Control deny all
-   cache_peer www.ara65.centos.com parent 80 0 no-query originserver

-   # Tamaño de memoria caché
-   cache_mem 256 MB

-   # Especificar nombre de servidor
-   visible_hostname prox.ara65.centos.com

-   # No mostrar direcciones IP
```

- `forwarded_for off`

Una vez configurado el fichero debemos instalar, en el caso de no tenerlo ya instalado, las “http-tools” con las que crearemos el fichero que utilizaremos para la autentificación web.

Creamos el fichero “/etc/squid/.htpasswd” con la orden “htpasswd -c /etc/squid/.htpasswd user”. Se nos pedirá que introduzcamos tanto usuario como contraseña y se crea automáticamente el archivo.

Debemos asegurarnos también de que el puerto 8080 está abierto.

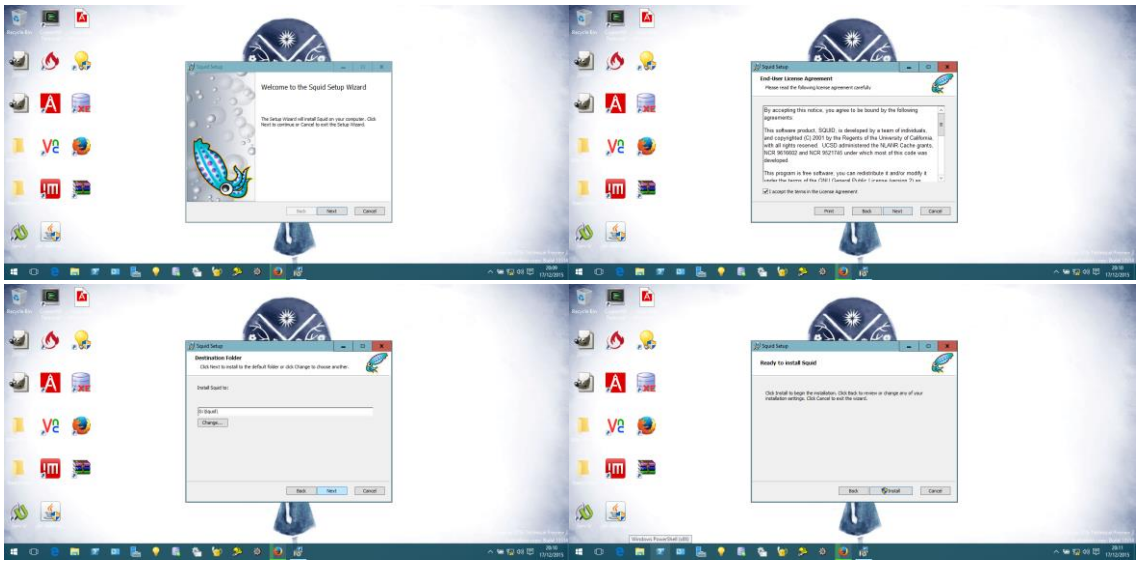
Una vez terminada la instalación y configuración habilitaremos el servicio con “systemctl enable squid” y lo iniciaremos “systemctl start squid”.

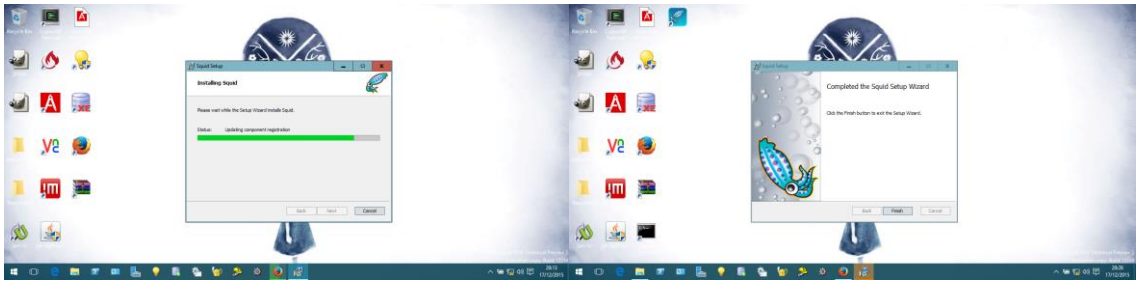
Para la instalación en FreeBSD utilizaremos el comando “pkg install squid” y configuraremos del mismo modo que en CentOS, sin embargo, los directorios de los archivos serán “/usr/local/etc/squid” en vez de “/etc/squid” y “/usr/local/libexec/squid” en vez de “/usr/lib64/squid” como en CentOS.

Construiremos la caché del servicio con la orden “squid -z” y deberemos añadir al archivo “/etc/rc.conf” la línea “`squid_enable=”YES”`” para que se inicie junto al equipo. Finalmente iniciamos el servicio con “service squid start”.

Como hemos comentado antes, para instalar Squid en Windows haremos uso de los binarios que nos proporciona la página oficial del proyecto.

Una vez descargados procederemos a la instalación.

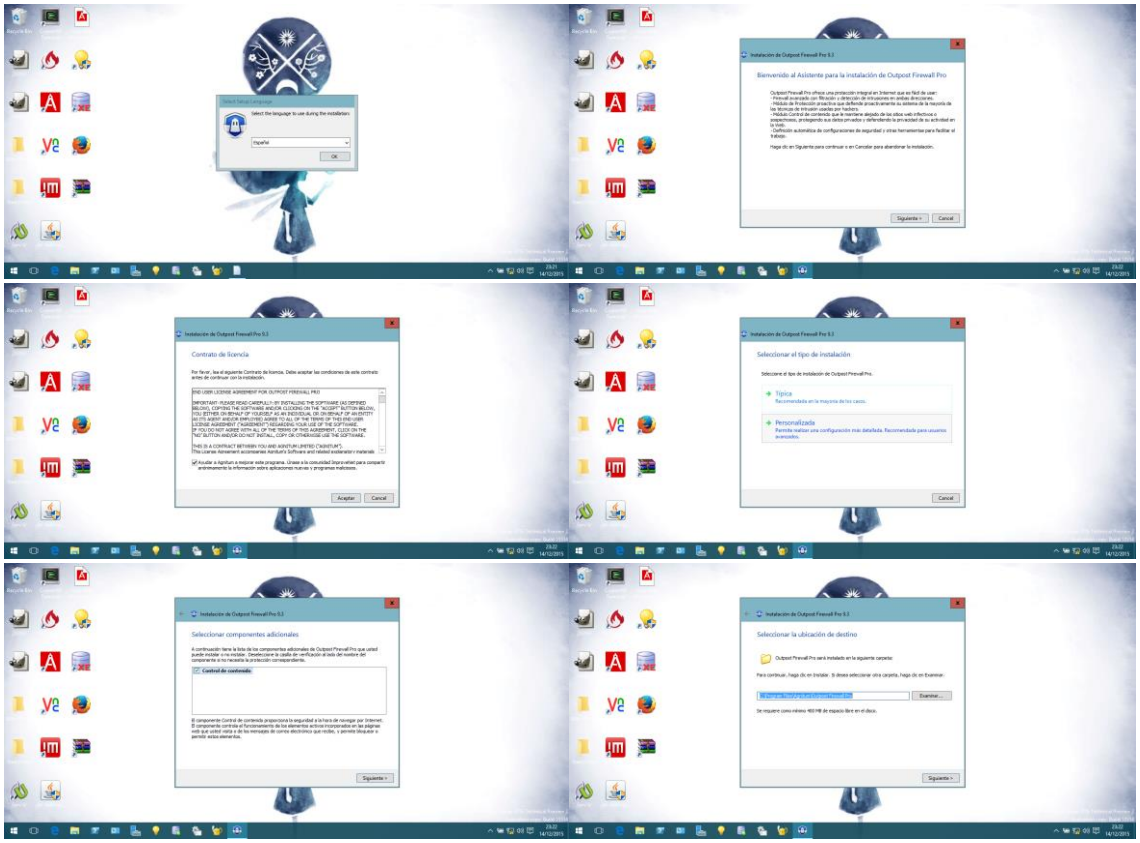


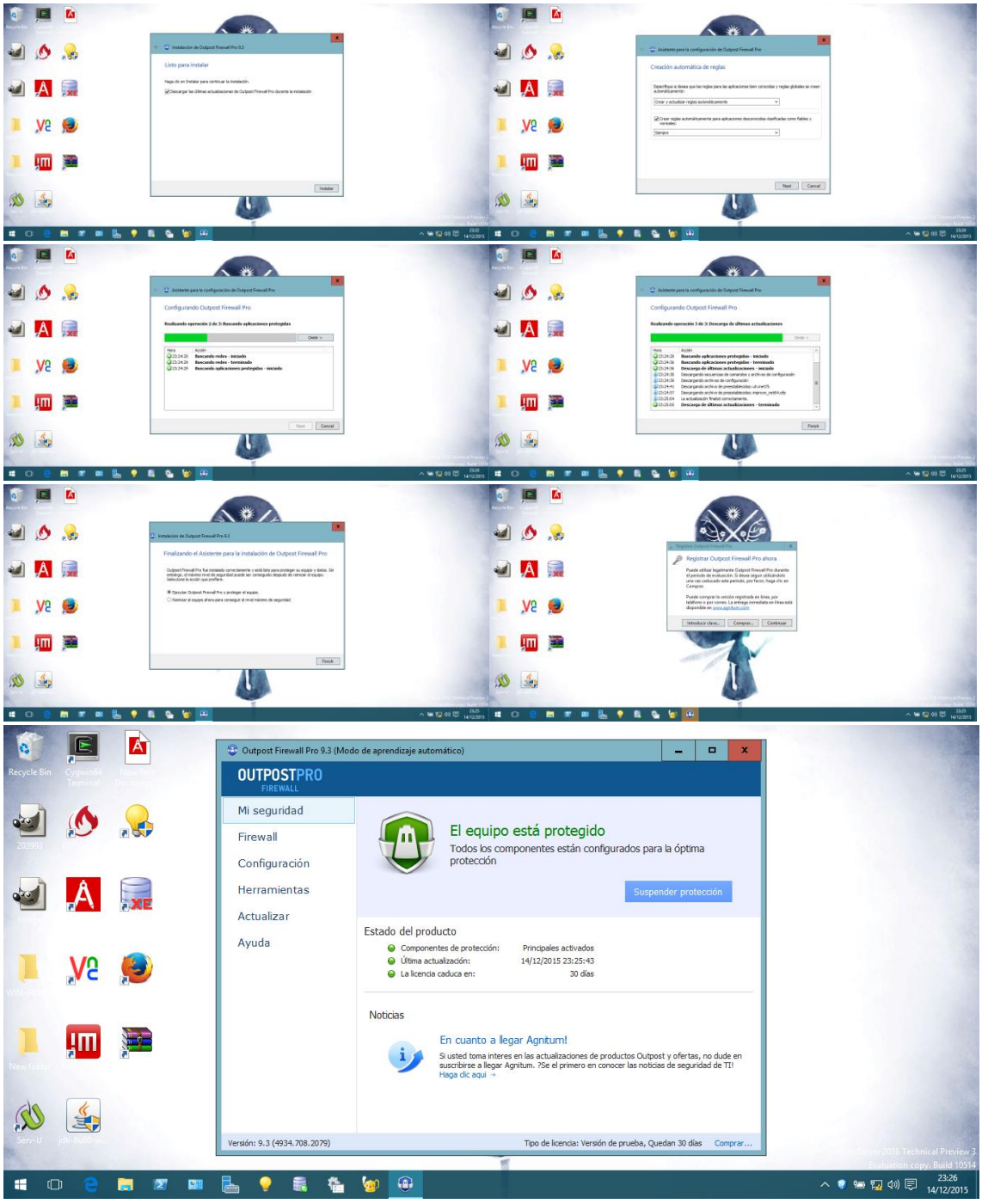


Terminada la instalación, abriremos el fichero de configuración "DirectorioInstalacionSquid/etc/squid.conf" y lo editaremos como se indica arriba, proporcionando un nuevo nombre para el proxy.

7. Rutado, Firewall y VPN

Para Windows hemos elegido instalar Outpost firewall pro, en su version de prueba de 30 dias, la cual podemos descargar desde la página oficial del proyecto.
Una vez descargado el instalador, simplemente debemos ejecutarlo:





Para instalar VPN en CentOS instalaremos el paquete OpenVPN desde epel con “yum -y install openvpn easy-rsa net-tools bridge-utils”.

Una vez instalado crearemos los certificados CA. Para ello nos dirigimos a la carpeta “/usr/share/easy-rsa/2.0” y editamos el archivo “vars” para que quede de la siguiente manera:

- `export KEY_COUNTRY="ES"`

```
- export KEY_PROVINCE="Alicante"
- export KEY_CITY="Elda"
- export KEY_ORG="Universidad de Alicante"
- export KEY_EMAIL="ara65@ara65.centos.com"
- export KEY_OU="Universidad de Alicante"
- export KEY_NAME="Server-CA"
```

A continuación ejecutamos los siguientes comandos para generar las claves necesarias:

```
- source ./vars
- ./clean-all
- ./build-ca
- ./build-key-server server
- ./build-dh
- ./build-key client01
```

Una vez ejecutados los comandos "./build-ca","./build-key-server server" y "./build-key client001"se deberá pulsar enter hasta terminar de generar el certificado y aceptar, en caso de ser necesario, que los cambios se guarden.

Unz vez generados los certificados necesarios, debemos configurar OpenVPN server para que funcione. Para ello ejecutamos los siguientes comados:

```
- cp -pR /usr/share/easy-rsa/2.0/keys /etc/openvpn/keys
- cp /usr/share/doc/openvpn-*/sample/sample-config-files/server.conf
  /etc/openvpn/
```

Tras esto editamos el archivo "/etc/openvpn/server.conf":

```
- port 1194    #Cambiar en caso de que sea necesario (línea 32)
- proto tcp    #Descomentar (línea 35)
- ;proto udp   #Comentar
- dev tap0     #Cambiar, ya que tap utiliza el modo bridge (línea 52)
- :dev tun     #Comentar
- ##Cambiar los paths (línea 78)
- ca keys/ca.crt
- cert keys/server.crt
- key keys/server.key
- dh keys/dh2048.pem        #(línea 85)
- ;server 10.8.0.0 255.255.255.0  #Comentar (línea 101 )
- #Descomentar y cambiar "server-bridge <ipVPNServer> <Máscara> <Rango
  de IP para el cliente>" (línea 120)
- server-bridge      192.168.122.4 255.255.255.0 192.168.122.1
  192.168.122.199
- keepalive 10 120   #Configuración del keepalive (línea 231)
```

```
- comp-lzo           #Habilitar compresión
- persist-key        #Habilitar opciones persistentes
- persist-tun
- log /var/log/openvpn.log  #Descomentar y especificar la ruta de los
  logs
- log-append /var/log/openvpn.log
- verb 3             #Nivel de debug (del 0 al 9, donde 9 significa
  nivel de debug)(línea 299)
```

A continuación ejecutamos los siguientes 3 comandos:

```
- cp /usr/share/doc/openvpn-*/sample/sample-scripts/bridge-start
  /etc/openvpn/openvpn-startup
- cp /usr/share/doc/openvpn-*/sample/sample-scripts/bridge-stop
  /etc/openvpn/openvpn-shutdown
- chmod 755 /etc/openvpn/openvpn-startup /etc/openvpn/openvpn-shutdown
```

Editamos el archivo "/var/openvpn/openvpn-startup":

```
- #Cambiar líneas si necesario (líneas 17-20)
- eth="enp0s8" #Cambiar si necesario
- eth_ip="192.168.122.4"    #IP para la interfaz de puente
- eth_netmask="255.255.255.0" #Máscara de subred
- eth_broadcast="192.168.122.255" #Dirección de broadcast
- #Añadir al final del archivo para definir la puerta de enlace por
  defecto
- eth_gw="192.168.122.1"
- route add default gw $eth_gw
```

Por último ejecutamos el comando "`cp /usr/lib/systemd/system/openvpn@.service /usr/lib/systemd/system/openvpn-bridge.service`" y editamos el archivo "`/usr/lib/systemd/system/openvpn-bridge.service`" para que quede de la siguiente manera:

```
- [Service]
- PrivateTmp=true
- Type=forking
- PIDFile=/var/run/openvpn/openvpn.pid
- ExecStartPre=/bin/echo 1 > /proc/sys/net/ipv4/ip_forward
- ExecStartPre=/etc/openvpn/openvpn-startup
- ExecStart=/usr/sbin/openvpn --daemon --writepid
  /var/run/openvpn/openvpn.pid --cd /etc/openvpn/ --config server.conf
- ExecStopPost=/etc/openvpn/openvpn-shutdown
- ExecStopPost=/bin/echo 0 > /proc/sys/net/ipv4/ip_forward
```

Una vez hecho esto iniciamos el servicio “openvpn-bridge” y lo habilitamos para que se inicie con el servidor con “systemctl start openvpn-bridge” y “systemctl enable openvpn-bridge”.

Para hacer el ruteado en FreeBSD debemos configurar varias cosas. Empezamos modificando el archivo “/etc/rc.conf”:

```
# Conexión WAN
ifconfig_xl0="inet 10.0.0.2 netmask 255.255.255.0"
#  If you are using dhcp for WAN connection use ifconfig_xl0="dhcp"

# Conexión LAN
ifconfig_xl1="inet 192.168.0.1 netmask 255.255.255.0"

# Puerta de enlace por defecto
defaultrouter="10.0.0.1" # Establecer la puerta de enlace por defecto

# Habilitar el reenvio de IP
gateway_enable="YES"

# Hostname
hostname="ara65.free.com"
```

Debes reemplazar xl0, xl1 con el dispositivo correcto para tu tarjeta, igual que con las direcciones correctas.
El router por defecto es necesario únicamente si no estás utilizando DHCP para la conexión WAN.

Si configuraste la red durante la instalación, algunas líneas a cerca de las tarjetas de red pueden estar ya presentes. Revisa el archivo “/etc/rc.conf” antes de añadir ninguna línea.

Necesitarás también editar el archivo “/etc/hosts” para añadir los nombres y las direcciones IP de las diferentes máquinas de la LAN, en caso de que no estén todavía ahí.

```
127.0.0.1        localhost localhost.my.domain
192.168.0.1    freebsdrouter.my.domain freebsdrouter
```

Establever el DNS en “/etc/resolv.conf”:

```
nameserver    10.0.0.1
```

Necesitarás configurar manualmente el nombre del servidor únicamente si no estás utilizando DHCP para una conexión a WAN.

Una vez hecho esto, configuraremos el DHCP editando el archivo "/usr/local/etc/dhcpd.conf":

```
# Nombre de servidor
option domain-name-servers 10.0.0.1;
# lease time
default-lease-time 600;
max-lease-time 7200;

# Si este servidor DHCP es el DHCP oficial de la red local, la
directiva authoritative debe estar descomentada.
authoritative;

# Actualización del esquema ad-hoc DNS establecido a "none" para
deshabilitar la actualización dinámica del DNS.
ddns-update-style none;

# Utiliza esto para enviar los mensajes de log de DHCP a un fichero
diferente.
# Deberás de hackear "syslog.conf" para completar la redirección
log-facility local7;

# Declaración de subred
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.100 192.168.0.200;
option routers 192.168.0.1;
option subnet-mask 255.255.255.0;
}
```

Habilita DHCP para iniciarse con el equipo modificando "/etc/rc.conf".

```
dhcpd_enable=YES
dhcpd_ifaces="xl1" # Tarjeta LAN
```

8. Monitorización de servicios

El software de monitorización Nagios, trabaja sobre los protocolos http y php, por lo que antes de instalar Nagios debemos tener instalados y configurados correctamente estos servicios. Además, en algunos sistemas nagios es descargado desde el repositorio EPEL por lo que tenemos que tenerlo instalado y activo.

Para instalar Nagios y sus plugins en CentOS, utilizaremos la orden "yum -y gd gd-devel gcc glibc glibc-common wget".

Una vez descargados los complementos necesarios, creamos el usuario nagios y el grupo nagcmd del siguiente modo:

Para el usuario

- useradd -m nagios
- passwd nagios

Para el grupo

- groupadd nagcmd
- usermod -a -G nagcmd nagios
- usermod -a -G nagcmd apache

Ahora descargamos las últimas versiones de nagios y de sus plugins con las órdenes:

- wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.1.1.tar.gz
- wget http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz

Una vez descargado nagios, lo descomprimimos con la órden:

- tar xzf nagios-4.1.1.tar.gz

Una vez descompreso, entraremos a la carpeta, le daremos permisos y procederemos a la instalación:

- cd nagios-4.1.1
- ./configure --with-command-group=nagcmd
- make all
- make install
- make install-init
- make install-config
- make install-commandmode
- make install-webconf

Una vez compilados e instalados todos los archivos necesarios, crearemos un usuario para conectarnos a la interfaz web y reiniciaremos el servidor web:

- htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
- systemctl restart httpd

Ahora volveremos al directorio donde descargamos los plugins para descomprimirlos, compilarlos e instalarlos:

- tar xzf nagios-plugins-2.1.1.tar.gz
- cd nagios-plugins-2.1.1
- ./configure --with-nagios-user=nagios --with-nagios-group=nagios
- make
- make install

Editamos el archivo “/usr/local/nagios/etc/objects/contacts.cfg” e indicamos la dirección de correo del administrador.

Editamos, también, el archivo “/etc/httpd/conf.d/nagios.conf” y editamos lo siguiente:

```
- # Order allow, deny      #Comentar
- # Allow from all         #Comentar

- #Descomentar las siguientes líneas y añadir la red desde la que se
  podrá conectar a nagios
- Order deny,allow
- Deny from all
- Allow from 127.0.0.1 192.168.122.0/24
```

Ahora reiniciamos de nuevo el servidor web con “systemctl restart httpd” y comprobamos que tenemos bien la configuración de nagios con “/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg”.

Una vez hecho esto solo nos falta iniciar el servicio con “service nagios start”, habilitarlo para que se inicie con el equipo “chkconfig --add nagios” y “chkconfig nagios on”.

Para la instalación de Nagios en FreeBSD necesitaremos instalarlo desde ports, por lo que segiremos los siguientes pasos:

```
- cd /usr/ports/net-mgmt/nagios
- make install clean
- # En caso de error por tener una librería más nueva instalada, hacer
  “make reinstall”
```

Aceptamos la instalación por defecto y debemos asegurarnos de seleccionar “NETSNMP” para que se creen tanto el grupo como el usuario de nagios.

Añadimos la línea “nagios_enable=”YES”” al archivo “/etc/rc.conf” y nos dirigimos al directorio “/usr/local/etc/nagios” para configurar el servicio.

Comenzaremos por copiar los archivos base de la configuración:

```
- cd /usr/local/etc/nagios
- cp cgi.cfg-sample cgi.cfg
- cp nagios.cfg-sample nagios.cfg
- cp resource.cfg-sample resource.cfg
```

De igual modo, copiaremos los ficheros base de la configuración correspondientes dirigiéndonos a la carpeta “/usr/local/etc/nagios/objects/”:

```
- cd /usr/local/etc/nagios/objects/
- cp commands.cfg-sample commands.cfg
- cp contacts.cfg-sample contacts.cfg
- cp localhost.cfg-sample localhost.cfg
```

- cp printer.cfg-sample printer.cfg
- cp switch.cfg-sample switch.cfg
- cp templates.cfg-sample templates.cfg
- cp timeperiods.cfg-sample timeperiods.cfg

Una vez todo en su sitio, comprobamos que la configuración es correcta con "nagios -v /usr/local/etc/nagios/nagios.cfg".

A continuación creamos un usuario y contraseña para la interfaz web de nagios con "htpasswd -c /usr/local/etc/nagios/htpasswd.users nagiosadmin".

Añadimos nagios en nuestra configuración de apache, modificando el archivo "/usr/local/etc/apache24/httpd.conf":

```
ScriptAlias /nagios/cgi-bin/ /usr/local/www/nagios/cgi-bin/

Alias /nagios /usr/local/www/nagios/
<Directory /usr/local/www/nagios>
      Options None
      AllowOverride None
      Order allow,deny
      Allow from all
      AuthName "Nagios Access"
      AuthType Basic
      AuthUSerFile /usr/local/etc/nagios/htpasswd.users
      Require valid-user
</Directory>
<Directory /usr/local/www/nagios/cgi-bin>
      Options ExecCGI
      AllowOverride None
      Order allow,deny
      Allow from all
      AuthName "Nagios Access"
      AuthType Basic
      AuthUSerFile /usr/local/etc/nagios/htpasswd.users
      Require valid-user
</Directory>
```

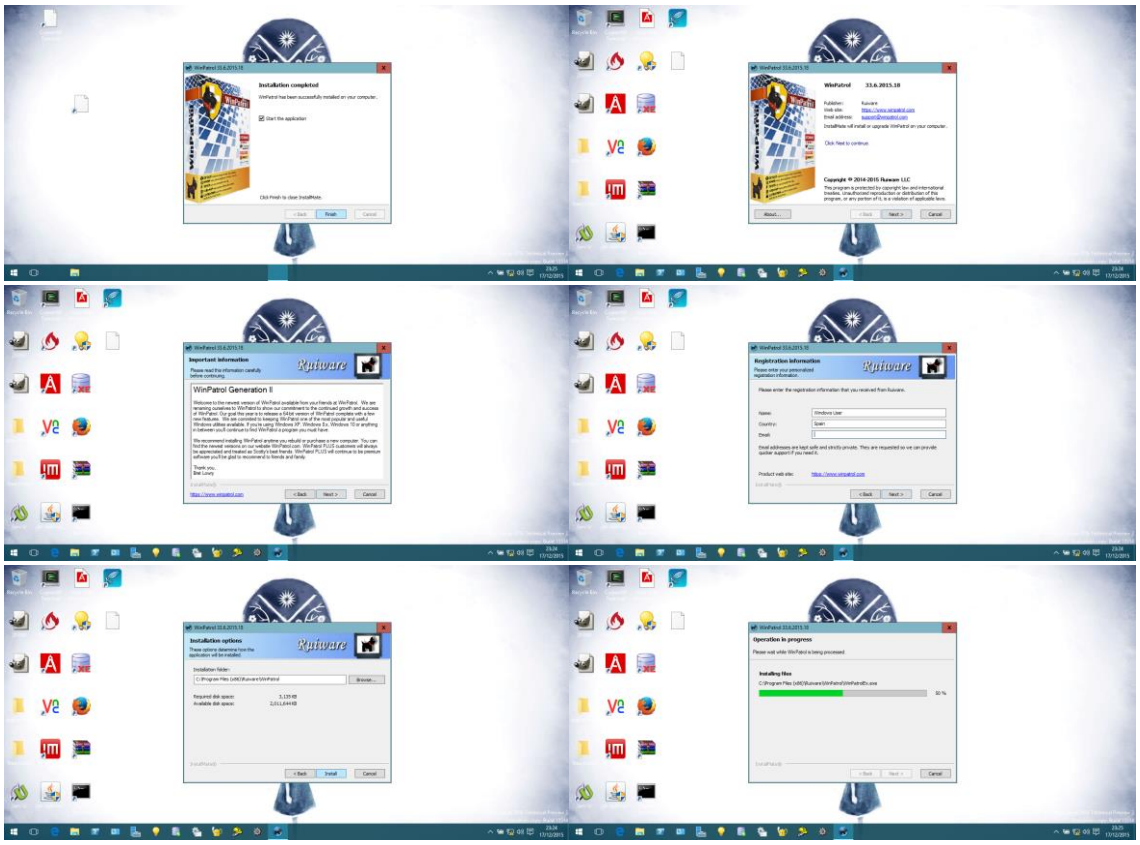
Por último reiniciamos el servidor web con "service apache24 restart" e iniciamos nagios con "service nagios start".

En el caso de que descargue los ".cgi" en vez de abrirlos, deberemos ir al archivo "/usr/local/etc/apache24/httpd.conf" y descomentar las líneas 160 y 163:

- LoadModule cgid_module libexec/apache24/mod_cgid.so
- LoadModule cgi_module libexec/apache24/mod_cgi.so

Después de guardar el archivo, reiniciamos el servidor web y ya nos abrirá los archivos.

Para Windows, utilizaremos WinPatrol, que nos provee de una interfaz local para monitorizar los servicios activos. Lo descargaremos desde su página oficial y lo instalaremos.



9. Ley de protección de datos.

La Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, (LOPD), es una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar. Fue aprobada en las Cortes españolas el 13 de diciembre de 1999. Esta ley se desarrolla fundamentándose en el artículo 18 de la constitución española de 1978, sobre el derecho a la intimidad familiar y personal y el secreto de las comunicaciones.

Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los crean o tratan.

Esta ley afecta a todos los datos que hacen referencia a personas físicas registradas sobre cualquier soporte, informático o no. Quedan excluidas de esta normativa aquellos datos recogidos para uso doméstico, las materias clasificadas del estado y aquellos ficheros que recogen datos sobre Terrorismo y otras formas de delincuencia organizada (no simple delincuencia).

A partir de esta ley se creó la Agencia Española de Protección de Datos, de ámbito estatal que vela por el cumplimiento de esta normativa.