

14-12-2017

# HITO 3

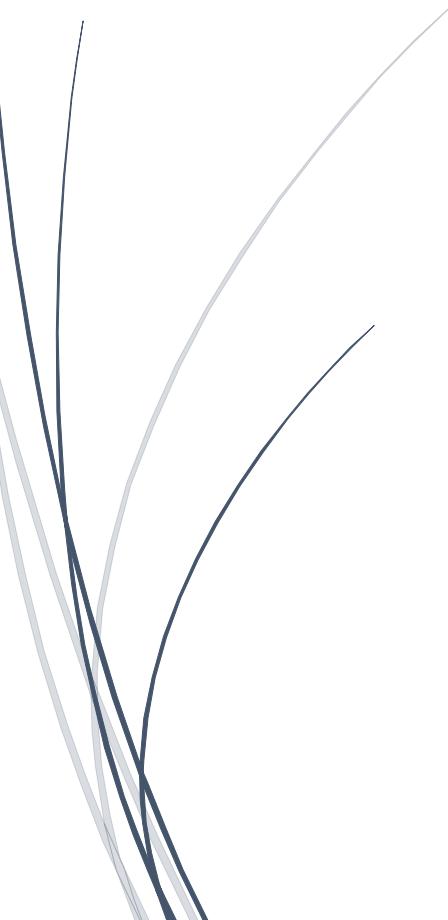
# ASORC

Realización de los siguientes  
**sistemas operativos**

CentOS 7 (**6 puntos**)

Freebsd (**7 puntos**)

Windows Server (**6 puntos**)



JAVIER RIVILLA ARREDONDO  
53247378D

CentOS 7 (6 puntos) ; Freebsd 11 (7 puntos); Windows server 2016 (6 puntos)

<b>CentOS 7</b> .....	1
Antes de comenzar.....	1
Servidor Web (Dominios virtuales).....	2
Servidor Base de datos (PostgreSQL) .....	5
Instant Messaging (Jabber).....	8
Servidor Proxy (Squid).....	13
Backup (Sincronización, absolutos, incremental) .....	17
Raid .....	19
<b>Freebsd</b> .....	21
Antes de comenzar.....	21
Servidor Web (Dominios virtuales).....	21
Servidor Base de datos (MySQL).....	23
Instant Messaging (Jabber).....	24
Servidor Proxy (Squid).....	26
Backup (Sincronización, absolutos, incremental) .....	29
Monitorización de servicios (Nagios) .....	29
Firewall.....	31
<b>Windows Server 2016</b> .....	31
Antes de comenzar.....	31
Servidor Web (Dominios virtuales).....	31
Instant Messaging (Jabber).....	33
Servidor Proxy (Squid).....	36
Backup (Copias de seguridad) .....	38
Raid .....	39
Monitorización de servicios (Spiceworks) .....	42

# CentOS 7

## Antes de comenzar

- Desactivaremos el firewall: `systemctl stop firewalld && systemctl disable firewalld`.

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# systemctl stop firewalld
[root@programador Programador]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor p
   reset: enabled)
     Active: inactive (dead)
       Docs: man:firewalld(1)
[root@programador Programador]#
```

- Desactivaremos el *SeLinux*: nano /etc/sysconfig/selinux

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1          Fichero: /etc/sysconfig/selinux

#
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=disable
# SELINUXTYPE= can take one of three two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes are pr$ 
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

## Servidor Web (Dominios virtuales)

Para esto tendremos que tener bien configurado el DNS del hito 2.

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# nano /etc/sysconfig/selinux
[root@programador Programador]# nslookup 192.168.56.101
Server:      192.168.56.101
Address:    192.168.56.101#53

101.56.168.192.in-addr.arpa      name = programador.cts-asorc.com.56.168.192.in-a
ddr.arpa.

[root@programador Programador]# nslookup cts-asorc.com
Server:      192.168.56.101
Address:    192.168.56.101#53

Name:  cts-asorc.com
Address: 192.168.56.120
Name:  cts-asorc.com
Address: 192.168.56.101
```

Para este hito yo añadí una más dentro del archivo: nano /etc/named.conf y configurando las nuevas zonas.

- Dominios: **cts-asorc.com** && **cts-asorc.es**

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# nslookup cts-asorc.es
Server:          192.168.56.101
Address:         192.168.56.101#53

Name:   cts-asorc.es
Address: 192.168.56.101
Name:   cts-asorc.es
Address: 192.168.56.120
```

- Instalamos el paquete:
  - **yum install httpd**
- Generamos la estructura de los directorios donde irán los contenidos web de cada dominio.

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1           Fichero: hito3.txt           Modificado
mkdir -p /var/www/cts-asorc.com/public_html/
mkdir -p /var/www/cts-asorc.es/public_html/
chown -R Programador:Programador /var/www/cts-asorc.com/public_html/
chown -R Programador:Programador /var/www/cts-asorc.es/public_html/
chmod -R 755 /var/www
```

Tecleando los pasos anteriores en la terminal.

Ahora dentro de cada dominio, instalaremos los correspondientes CMS. Para ello usaremos:

- Dentro de /var/www/cts-asorc.com/public\_html: WordPress.
- Dentro de /var/www/cts-asorc.es/public\_html: Joomla.

Creamos las carpetas en donde irán las configuraciones de los dominios y cuáles de ellos estarán activados (el segundo será un subconjunto de enlaces simbólicos hacia los archivos del primero; esto es para una mejor administración).

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1           Fichero: hito3.txt           Modificado
mkdir -p /etc/httpd/sites-available/
mkdir -p /etc/httpd/sites-enabled/
```

Luego nos vamos al fichero: **/etc/httpd/conf/httpd.conf**

The screenshot shows a terminal window titled "Programador@programador:/home/Programador". The menu bar includes Archivo, Editar, Ver, Buscar, Terminal, and Ayuda. The title bar says "GNU nano 2.3.1 Fichero: /etc/httpd/conf/httpd.conf". The content of the file is as follows:

```
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
IncludeOptional sites-enabled/*.conf
```

Creamos el fichero de configuración de **sites-available** para **cts-asorc.com**:

```
[root@programador Programador]# nano /etc/httpd/conf/httpd.conf
[root@programador Programador]# cat /etc/httpd/sites-available/cts-asorc.com.conf
<VirtualHost *:80>
ServerName cts-asorc.com
ServerAlias www.cts-asorc.com
DocumentRoot /var/www/cts-asorc.com/public_html
ErrorLog /var/www/cts-asorc.com/error.log
CustomLog /var/www/cts-asorc.com/requests.log combined
</VirtualHost>

[root@programador Programador]#
```

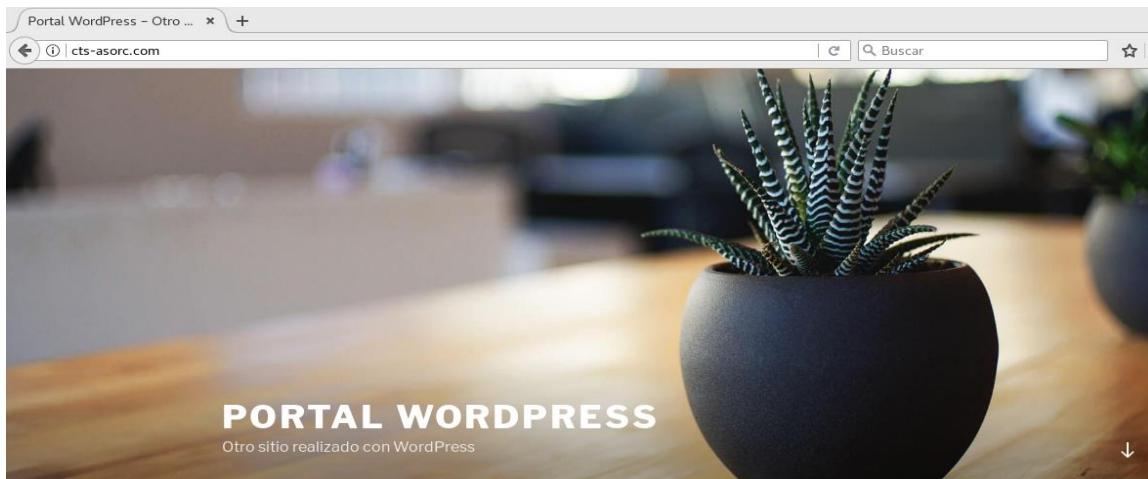
Y lo mismo para **cts-asorc.es**.

- Agregamos los enlaces simbólicos para el directorio de **sites-enabled**:
  - `In -s /etc/httpd/sites-available/cts-asorc.com.conf /etc/httpd/sites-enabled/cts-asorc.com.conf`
  - `In -s /etc/httpd/sites-available/cts-asorc.es.conf /etc/httpd/sites-enabled/cts-asorc.es.conf`
- Iniciamos el servicio y hacemos que cargue el sistema:
  - `Systemctl enable httpd`
  - `Systemctl start httpd`

Desde el host Linux:

Firefox → Arriba en el navegador ponemos nuestro dominio → Si no conecta fijarse en el **/etc/resolv.conf**.

- Cts-asorc.com → WordPress (*esto es el CMS*).



- Cts-asorc.es → Joomla (*esto es el CMS*)

The screenshot shows a Joomla website with the following details:

- Title:** Portal Joomla ES
- Page:** Home
- Main Menu:** Home
- Login Form:** Includes fields for Username and Password, a Remember Me checkbox, and a Log In button. It also includes links for 'Forgot your username?' and 'Forgot your password?'

Instalar el WordPress/Joomla en el dominio no lo he explicado, simplemente el mecanismo, pero me guiado por esta página:

<https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-centos-7>

## Servidor Base de datos (PostgreSQL)

Para CentOS yo he utilizado PostgreSQL, para ello haremos los siguientes pasos:

- Instalamos el paquete que corresponde a PostgreSQL:

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[Programador@programador ~]$ su
[root@programador Programador]# yum install postgresql-server postgresql-contrib
```

- Creamos la base de datos inicial para la propia configuración del servicio:

```

Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# postgres
postgres          postgresql-check-db-dir postgresql-setup
[root@programador Programador]# postgresql-setup initdb

```

Una vez hacemos esto, nos vamos al archivo de configuración

**/var/lib/pgsql/data/pg\_hba.conf** para habilitar conexiones remotas a través de password:

```

# "local" is for Unix domain socket connections only
local  all      all                                     peer
# IPv4 local connections:
host   all      all      127.0.0.1/32               md5
# IPv6 local connections:
host   all      all      ::1/128                 md5
# IPv4 192.168.0.0/16 connections:
host   all      all      192.168.0.0/16            md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local  replication      postgres                  peer
#host   replication      postgres      127.0.0.1/32           ident

```

- Añadimos dentro del archivo de configuración **/var/lib/pgsql/data/postgresql.conf**

```

Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1      Fichero: /var/lib/pgsql/data/postgresql.conf

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for all
                                # (change requires restart)
#port = 5432                    # (change requires restart)
# Note: In RHEL/Fedora installations, you can't set the port number here;
# adjust it in the service file instead.
max_connections = 100           # (change requires restart)

```

- Iniciamos y habilitamos el servicio:
  - **Systemctl enable postgresql && systemctl start postgresql**

Ahora desde el usuario PostgreSQL vamos a crear el nuestro, para acceder desde el usuario programador:

```

Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# su postgres -
bash-4.2$ psql
could not change directory to "/home/Programador"
psql (9.2.23)
Type "help" for help.

postgres=# \password
Enter new password:
Enter it again:
postgres=# \q

```

Ahora una vez hemos establecido la contraseña creamos el usuario:

```
bash-4.2$ createuser --interactive
could not change directory to "/home/Programador"
Enter name of role to add: multimedia
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
bash-4.2$ createdb multimedia
could not change directory to "/home/Programador"
bash-4.2$ exit
exit
```

- Ahora accedemos al usuario PostgreSQL:

```
Programador@programador:~ - □ ×
Archivo Editar Ver Buscar Terminal Ayuda
[Programador@programador ~]$ psql
psql (9.2.23)
Digite <help> para obtener ayuda.

Programador=>
```

Para crear una table debemos de poner:

- Create table amigotes (  
    id serial primary key,  
    nombre varchar(20) not null,  
    edad smallint  
);

Ahora añadimos a la tabla unos cuantos atributos:

- Insert into amigotes (nombre, edad) values ('Javier', 20);
- Insert into amigotes (nombre, edad) values ('Francisco', 14);
- Insert into amigotes (nombre, edad) values ('Óscar', 26);
- Insert into amigotes (nombre, edad) values ('Ángel', 20);
- Insert into amigotes (nombre, edad) values ('Ronald', 21);

Y accedemos a la tabla con **select \* from amigotes;**

```
Programador=> select * from amigotes
Programador-> ;
 id | nombre   | edad
----+-----+-----
 1 | Javi     | 20
 2 | Francisco | 14
 3 | Oscar    | 26
 4 | Angel    | 20
 5 | Ronald   | 21
(5 filas)
```

Como el profesor nos pide que utilicemos un programa en PHP:

```
Programador@programador:/home/Programador - □ ×
Archivo Editar Ver Buscar Terminal Ayuda
[Programador@programador ~]$ su
Contraseña:
[root@programador Programador]# yum install php php-pqsql
```

- Con fines de desarrollo, modificamos la directiva **display\_errors** poniéndola a **on** en el archivo **/etc/php.ini** para que pueda mostrar todos los errores (incluso desde el html mostrado por el navegador) de código PHP.
- Creamos un script (con copiar el código y poner nuestra tabla sobra).

```
[root@programador Programador]# cat /var/www/cts-asorc.com/public_html/db.php

<?php
$conn_params = "host=192.168.56.101 port=5432 dbname=Programador user=Programador
password=rivilla2";
$conn = pg_connect($conn_params);
$result = pg_query($conn, "select * from amigotes");
echo("<table border=2>\n");
echo(" <tr>\n");
for($i = 0; $i < pg_num_fields($result); $i++) {
$col_name = pg_field_name($result, $i);
echo(" <th>$col_name</th>\n");
}
echo(" </tr>");
while($line = pg_fetch_array($result, null, PGSQL_ASSOC)) {
echo(" <tr>\n");
foreach($line as $col_value => $row_value) {
echo(" <td>$row_value</td>\n");
}
echo(" </tr>\n");
}
echo("</table>\n");
?>
```

Nos vamos desde el host → Firefox → [cts-asorc.com/db.php](http://cts-asorc.com/db.php)

<b>id</b>	<b>nombre</b>	<b>edad</b>
1	Javi	20
2	Francisco	14
3	Oscar	26
4	Angel	20
5	Ronald	21

Y comprobamos que funcionar correctamente.

## Instant Messaging (Jabber)

Lo primero de todo instalamos el paquete desde la página oficial en el escritorio, por ejemplo:

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# ls
backup Escritorio Imágenes openfire-3.9.3-1.i386.rpm pruebacups.pdf Vídeos
Descargas FTP Música OTROS Público
Documentos hito3.txt NFS Plantillas SAMBA
[root@programador Programador]# yum install openfire-3.9.3-1.i386.rpm
```

- También instalamos la librería C estándar para 64 bits.

```

Programador@programador:/home/Programador
- □ ×
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# ls
backup Escritorio Imágenes openfire-3.9.3-1.i386.rpm pruebacups.pdf Vídeos
Descargas FTP Música OTROS Público
Documentos hito3.txt NFS Plantillas SAMBA
[root@programador Programador]# yum install glibc.i686

```

- Iniciamos el servicio y lo habilitamos:
  - **Systemctl enable openfire && systemctl start openfire**

Accedemos al Firefox y ponemos: 192.168.56.101:9090

- Más general sería <Nuestra ip>:<puerto>

**Choose Language**

**Czech (cs\_CZ)**  
 **Deutsch (de)**  
 **English (en)**  
 **Español (es)**  
 **Français (fr)**  
 **Nederlands (nl)**  
 **Polski (pl\_PL)**  
 **Português Brasileiro (pt\_BR)**  
 **Русский (ru\_RU)**  
 **Slovenčina (sk)**  
 **中文 (简体) Simplified Chinese (zh\_CN)**

**Continue**

En el dominio podemos poner tanto **cts-asorc.com** como nuestra ip, en este caso he cogido una captura de Pavel, pero nuestra ip sería **192.168.56.101**.

Dominio:  (?)  
 Puerto de la Consola de Administración:  (?)  
 Puerto de la Consola de Administración Segura:  (?)  
 Property Encryption via: (?)  
 **Blowfish**  
 **AES**  
 Property Encryption Key:   
  
(?)

**Continuar**

Aquí más de lo mismo:

**Por defecto**

Almacenar usuarios y grupos en la base de datos de Openfire. Esta es la mejor opción para instalaciones simples.

**Only Hashed Passwords**

Store only non-reversible hashes of passwords in the database. This only supports PLAIN and SCRAM-SHA-1 capable clients.

**Servidor de Directorio (LDAP)**

Integrar con un servidor de directorio como ser Active Directory o OpenLDAP utilizando el protocolo LDAP. Usuarios y grupos van a ser almacenados en el directorio y tratados como de sólo-lectura.

**Continuar**

- En esta última página establecemos la contraseña del administrador, para configurar las cuentas en el openfire:

Correo Electrónico del Administrador:   
Una dirección de correo electrónico válida para la cuenta del administrador.

Nueva Contraseña:

Confirme la Contraseña:

**Saltar este paso** **Continuar**

La password que yo he utilizado es: **rivilia2** en todos los sitios.

- Nos logueamos en el **openfire**:



- Nos vamos a usuarios dentro de **Usuarios/Grupos**.

The screenshot shows a web-based administration interface for user management. At the top, there are two tabs: 'Servidor' and 'Usuarios/Grupos'. The 'Usuarios/Grupos' tab is highlighted with an orange background. Below it, there are two sub-tabs: 'Usuarios' and 'Grupos', separated by a dotted line. Under the 'Usuarios' tab, there is a yellow arrow pointing right next to the text 'Lista de Usuarios'. Below this, there are three other options: 'Crear Nuevo Usuario', 'Buscar Usuario', and 'Advanced User Search'.

- Y creamos el usuario que queramos en nuestro caso será “Prueba”.

The screenshot shows a 'Crear Nuevo Usuario' (Create New User) form. The form fields are as follows:

- Usuario: \*
- Nombre:
- Correo Electrónico:
- Contraseña: \*
- Confirmar Contraseña: \*
- ¿Es Administrador?  (Permite acceso de administrador a Openfire)

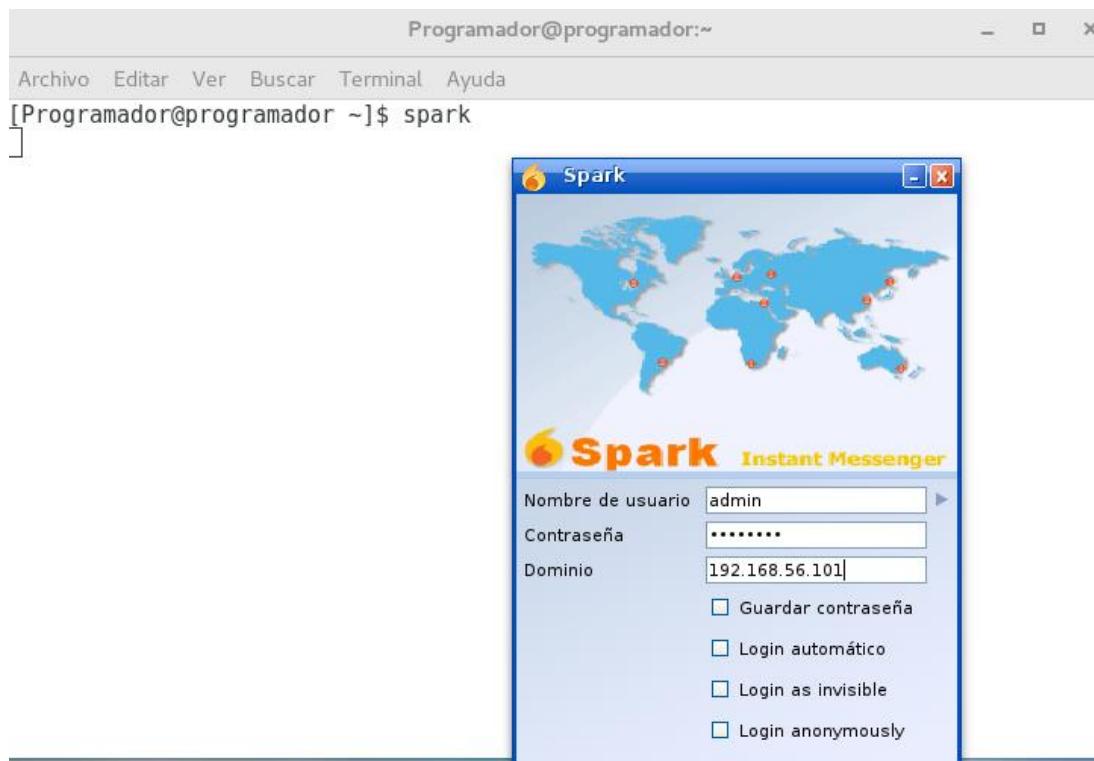
At the bottom of the form are three buttons: 'Crear Usuario', 'Crear y Crear Otro', and 'Cancelar'.

Y bueno muy importante:

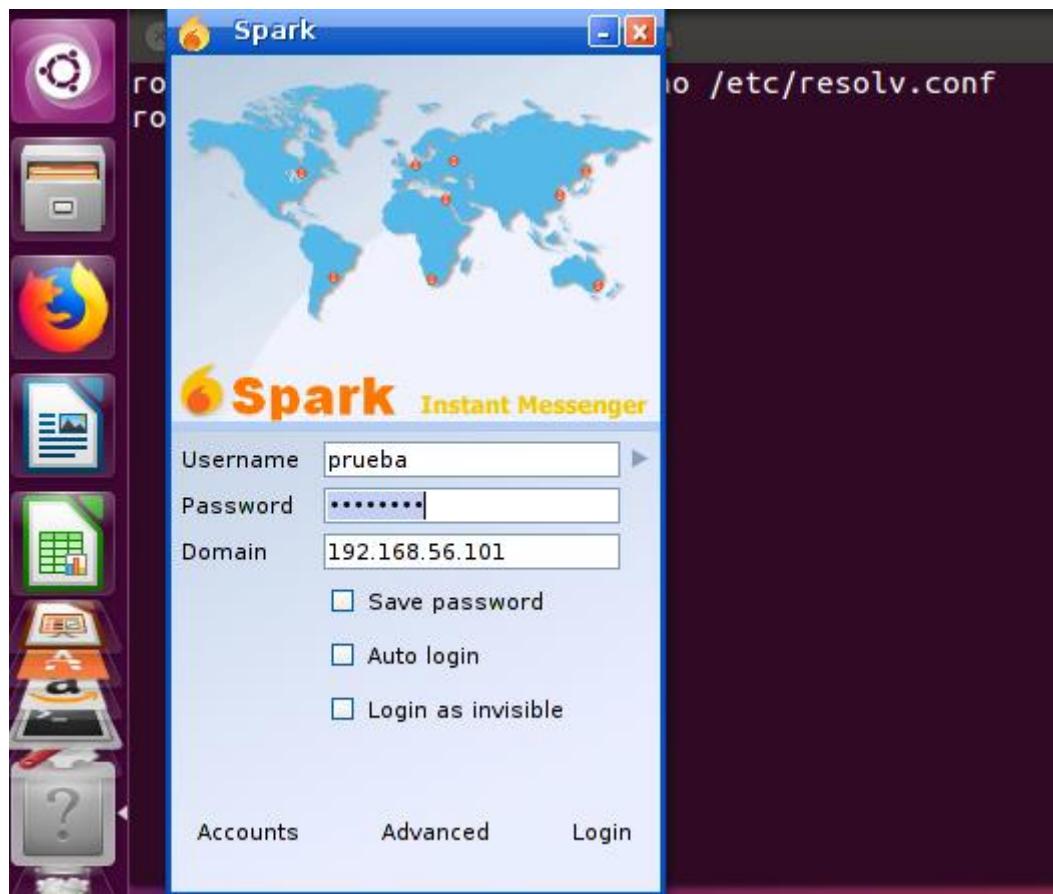
- Instalamos el cliente **Spark** en Centos7 y en el cliente Linux y accederemos a mandarnos mensajes.

Abrimos el **Spark** en los dos sistemas operativos:

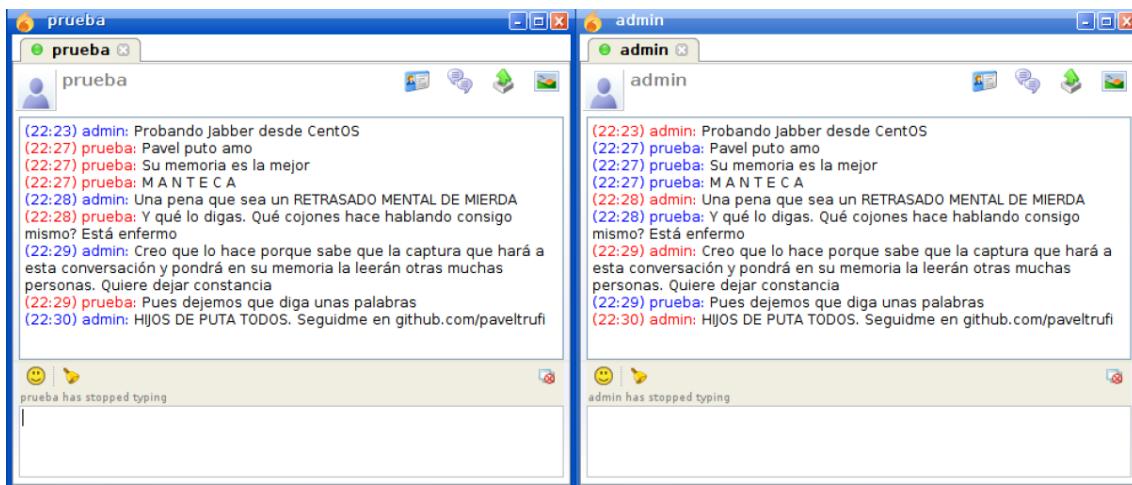
- En el CentOS se inicia poniendo en el terminal **Spark**.



Y en el Linux cliente buscamos el Spark en los programas y lo abrimos:



Establecemos conexión y comprobamos que funcionar correctamente:



## Servidor Proxy (Squid)

Para este servicio lo que haremos será primero instalar el correspondiente paquete y configurar los siguientes archivos de configuración.

- Instalamos el paquete:

```
Programador@programador:/home/Programador
-
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# yum -y install squid
```

- Creamos tres archivos de configuración dentro de **/etc/squid/**
  - Blocked\_sites, blocked\_images

En estos archivos vamos a bloquear las páginas que queramos en nuestro caso **www.marca.com**, las imágenes de ciertas extensiones por ejemplo png.

```
Programador@programador:/home/Programador
-
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1      Fichero: /etc/squid/blocked_images
.jpg$
.png$
```

```
Programador@programador:/home/Programador
-
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1      Fichero: /etc/squid/blocked_sites
www.marca.com
```

- Ahora modificaremos el archivo de configuración **/etc/squid/squid.conf**
  - También destacará la autenticación por usuario, es decir, cada vez que se ejecute el proxy nos pedirá logueo de usuario.

```

Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1 Fichero: /etc/squid/squid.conf

acl Safe_ports port 210      # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280      # http-mgmt
acl Safe_ports port 488      # gss-http
acl Safe_ports port 591      # filemaker
acl Safe_ports port 777      # multiling http
acl CONNECT method CONNECT
acl redlocal src 192.168.56.0/24 ←
acl blocked_sites url_regex "/etc/squid/blocked_sites" ←
acl blocked_images urlpath_regex "/etc/squid/blocked_images" ←

#auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/usuarios_$
acl password proxy_auth REQUIRED ←

#
# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports

```

Añadimos las líneas marcadas arriba, la primera y la última es para la autentificación de usuario y las dos del medio son para los directorios creados.

```

Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1 Fichero: /etc/squid/squid.conf

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# Only allow cachemgr access from localhost
http_access deny blocked_sites ←
http_access deny blocked_images ←
http_access allow password ←
http_access allow localhost manager ←
http_access deny manager

```

Y añadir las que faltan, pero esas de arriba son las más importantes.

- Una vez tenemos eso lo que tenemos que hacer es generar una password al usuario que queramos para ello teclearemos:

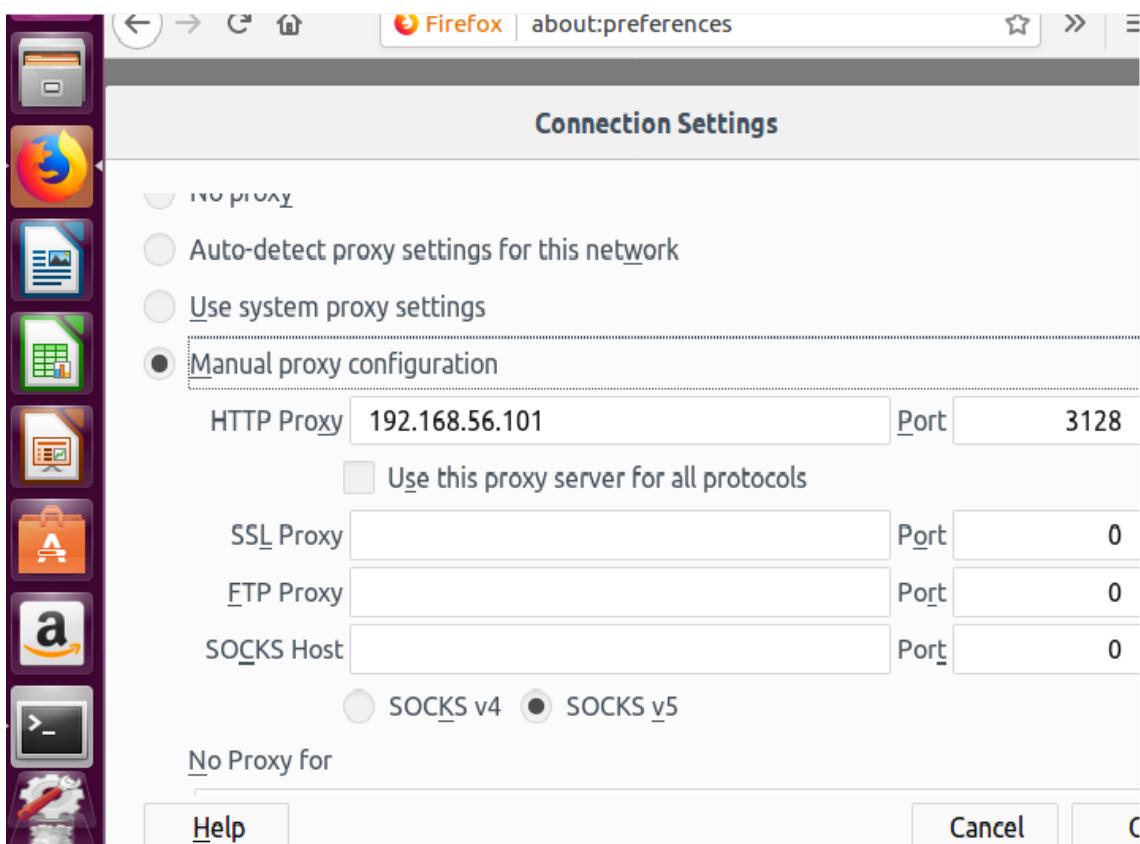
```
[root@programador squid]# ls
blocked_images  cachemgr.conf.default  mime.conf          squid.conf.default
blocked_sites   errorpage.css           mime.conf.default usuarios_permitidos
cachemgr.conf   errorpage.css.default  squid.conf
[root@programador squid]# htpasswd usuarios_permitidos programador
```

```
[root@programador squid]# cat usuarios_permitidos
programador:$apr1$SDiKHWA3$wncA3dTaMIBdWz0Ar9i90/
[root@programador squid]#
```

Establecemos y habilitamos el servicio:

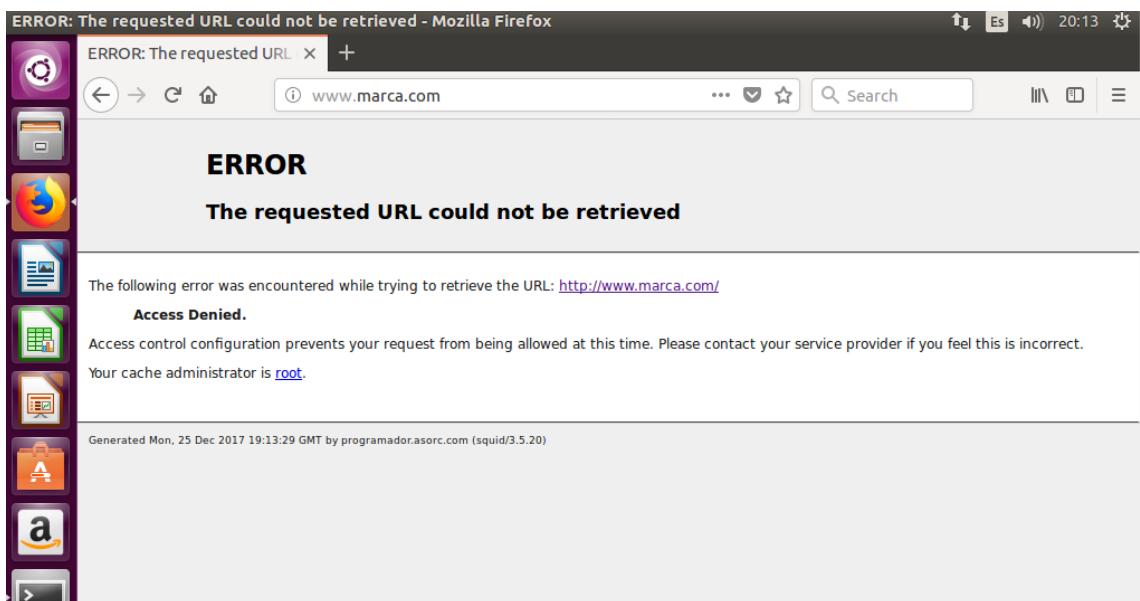
- **Systemctl enable squid && systemctl start squid**

- Desde el host, configuramos el Firefox para ello:
  - Firefox → Preferencias → Abajo del todo proxy



Una vez ponemos la ip de nuestro servidor comprobamos que el servicio funciona correctamente:

- Comprobamos la página **marca.com** como la bloquea nuestro proxy y otra página como Facebook no la bloquea.



ERROR: The requested URL [Facebook - Log In or Sign Up](#)

To help personalize content, tailor and measure ads, and provide a safer experience, we use cookies. By clicking or navigating the site, you agree to allow our collection of information on and off Facebook through cookies. Learn more, including about available controls: [Cookies Policy](#).

# facebook

Email or Phone      Password

Forgot account?

## Sign Up

It's free and always will be.

First name      Last name

Mobile number or email

New password

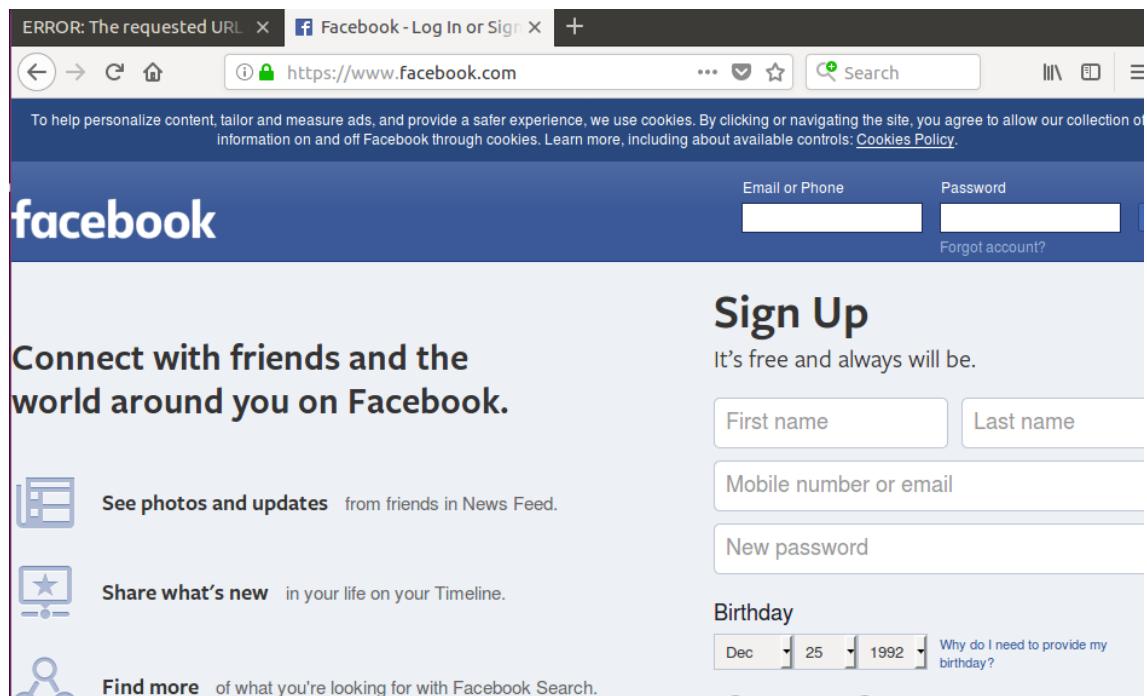
Birthday

Dec 25 1992 [Why do I need to provide my birthday?](#)

See photos and updates from friends in News Feed.

Share what's new in your life on your Timeline.

Find more of what you're looking for with Facebook Search.



- Comprobamos que pide el usuario y la contraseña:

Firefox Web Browser

New Tab

Search or enter address

Search the Web

TOP SITES

Authentication Required

The proxy moz-proxy://192.168.56.101:3128 is requesting a username and password. The site says: "Squid proxy-caching web server"

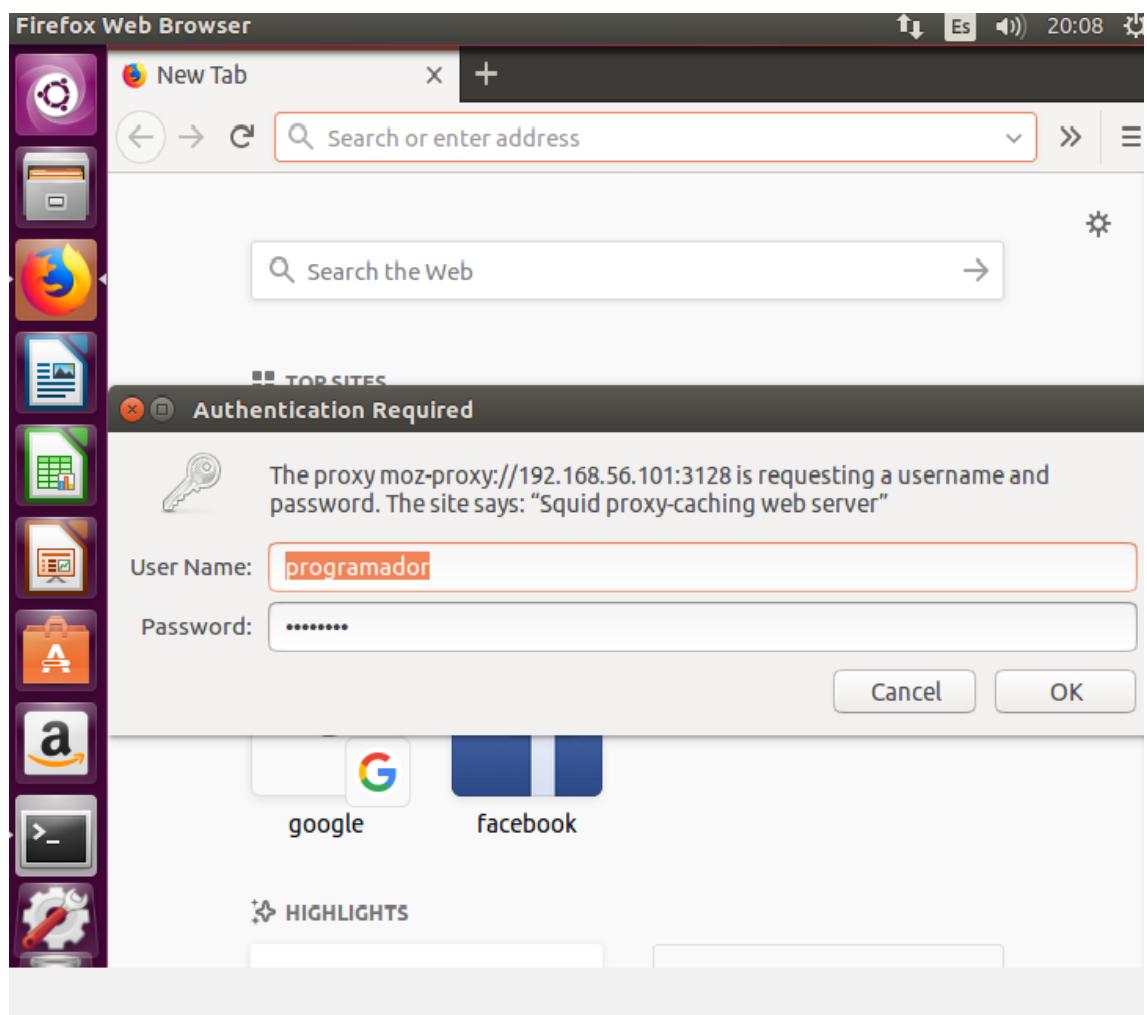
User Name:

Password:

Cancel      OK

google      facebook

HIGHLIGHTS



# Backup (Sincronización, absolutos, incremental)

- El paquete ya se encuentra instalado, pero por si acaso he utilizado **rsync**.

```
Programador@programador:/etc/squid
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador squid]# yum -y install rsync
```

Comento, este servicio desde mi punto de vista se puede hacer tanto en servidor, como en cliente, pero deberemos de enseñar que funcionar y comprendemos correctamente el servicio. Para ello deberemos de mostrar los siguientes pasos:

root@debian9:/home/programador/Escritorio# cat backup  
1 - Desde el cliente nos conectamos al servidor:

```
rsync -avz -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
```

2 - creamos dentro de la carpeta del servidor -> /home/Programador/backup/copia  
-> COMPLETO  
cd /home/Programador/backup/copia -> touch prueba.txt

3 - nos vamos al cliente y ejecutamos:

```
INCREMENTAL-> rsync -avr -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
```

4 - eliminamos el prueba.txt ----> SINCRONIZACIÓN

```
rsync -rtv --delete-delay Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
root@debian9:/home/programador/Escritorio#
```

- **Copia absoluta:** Es la primera que hacemos (desde un cliente Debian comprobamos):
  - **Rsync -avz -e ssh**  
[Programador@192.168.56.101:/home/Programador/backup](http://Programador@192.168.56.101:/home/Programador/backup)  
[/home/programador/backupServidor/](http://Programador@192.168.56.101:/home/programador/backupServidor/)

Debemos de tener en el servidor (CentOS 7) dentro de **/home/Programador/** la carpeta Backup.

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador backup]# cd
[root@programador ~]# cd /home/Programador/
[root@programador Programador]# ls
backup      FTP      NFS          pruebacups.pdf
Descargas     hito3.txt   openfire-3.9.3-1.i386.rpm    Público
Documentos    Imágenes   OTROS          SAMBA
Escritorio   Música    Plantillas    Vídeos
[root@programador Programador]#
```

Nos vamos al cliente Debian 9 Stable:

```

programador@debian9: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
programador@debian9: ~
programador@debian9: ~
root@debian9:/home/programador# rsync -avz -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
Enter passphrase for key '/root/.ssh/id_rsa':
receiving incremental file list
backup/
sent 15 bytes received 47 bytes 13.78 bytes/sec
total size is 0 speedup is 0.00
root@debian9:/home/programador#

```

- **Copia incremental:** Creamos un archivo en el servidor y hacemos el siguiente comando, se deberá “incrementar” los archivos dentro, es decir, se deberá de copiar en el cliente.
  - **Rsync -avz -e ssh**  
[Programador@192.168.56.101:/home/Programador/backup](#)  
[/home/programador/backupServidor/](#)

```

Debian Stable 9 (Aquí rdp Funciona) [Corriendo] - Oracle VM VirtualBox
CentOS7.17 (apache, db, proxy, spark) [Corriendo] - Oracle VM VirtualBox
Actividades Terminal lun 22:56
programador@debian9: ~
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
programador@debian9: ~
programador@debian9: ~
root@debian9:/home/programador# rsync -avz -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
Enter passphrase for key '/root/.ssh/id_rsa':
receiving incremental file list
backup/
backup/copia_seguridad/
backup/copia_seguridad/prueba.txt
sent 38 bytes received 137 bytes 50.00 bytes/sec
total size is 0 speedup is 0.00
root@debian9:/home/programador#

```

```

CentOS7.17 (apache, db, proxy, spark) [Corriendo] - Oracle VM VirtualBox
Aplicaciones Lugares Terminal lun 22:56
Programador@programador:/home/Programador/backup/copia_seguridad
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador backup]# cd /home/Programador/Programador
[root@programador Programador]# ls
backup  FTP  pruebacups.pdf
Descargas  hito3.txt  openfire-3.9.3-1.i386.rpm  Público
Documentos  Imágenes  OTROS  SAMBA
Escritorio  Música  Plantillas  Videos
[root@programador Programador]# cd backup/
[root@programador backup]# mkdir copia_seguridad
[root@programador backup]# ls
copia_seguridad
[root@programador backup]# cd copia_seguridad/
[root@programador copia_seguridad]# touch prueba.txt
[root@programador copia_seguridad]#

```

- **Copia sincronizada:** Simplemente tenemos que borrar el archivo de prueba de dentro de copia de seguridad y tecleamos el comando siguiente:
  - **rsync -rtv --delete-delay**  
[Programador@192.168.56.101:/home/Programador/backup](#)  
[/home/programador/backupServidor/](#)

```

Debian Stable 9 (Aquí rdp funciona) [Corriendo] - Oracle VM VirtualBox
CentOS7.17 (apache, db, proxy, spark) [Corriendo] - Oracle VM VirtualBox

[programador@debian9: ~]
[programador@debian9: ~]
root@debian9:/home/programador/Escritorio# cat backup
Prueba
imprimir cliente servidor
root@debian9:/home/programador/Escritorio# cat backup
1 - Desde el cliente nos conectamos al servidor:
rsync -avz -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor

2 - creamos dentro de la carpeta del servidor -> /home/Programador/backup/copia
-> COMPLETO
cd /home/Programador/backup/copia -> touch prueba.txt

3 - vamos al cliente y ejecutamos:
INCREMENTAL-> rsync -avr -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
root@debian9:/home/programador/Escritorio# 

[programador@programador:/home/Programador/backup/copia_seguridad]
[programador@programador:/home/Programador/backup/copia_seguridad]
[root@programador backup]# cd
[root@programador ~]# cd /home/Programador/
[root@programador Programador]# ls
backup  FTP  pruebacups.pdf
Descargas  hito3.txt  openfire-3.9.3-1.i386.rpm  Público
Documentos  Imágenes  OTROS  SAMBA
Escritorio  Música  Plantillas  Videos
[root@programador Programador]# cd backup/
[root@programador backup]# mkdir copia_seguridad
[root@programador backup]# ls
copia_seguridad
[root@programador backup]# cd copia_seguridad/
[root@programador copia_seguridad]# touch prueba.txt
[root@programador copia_seguridad]# rm -r prueba.txt
rm: borrar el fichero regular vacío «prueba.txt»? (s/n)
[root@programador copia_seguridad]# rm -r prueba.txt
rm: borrar el fichero regular vacío «prueba.txt»? (s/n) s
[root@programador copia_seguridad]# ls
[root@programador copia_seguridad]#

```

Y como podemos observar se borra correctamente:

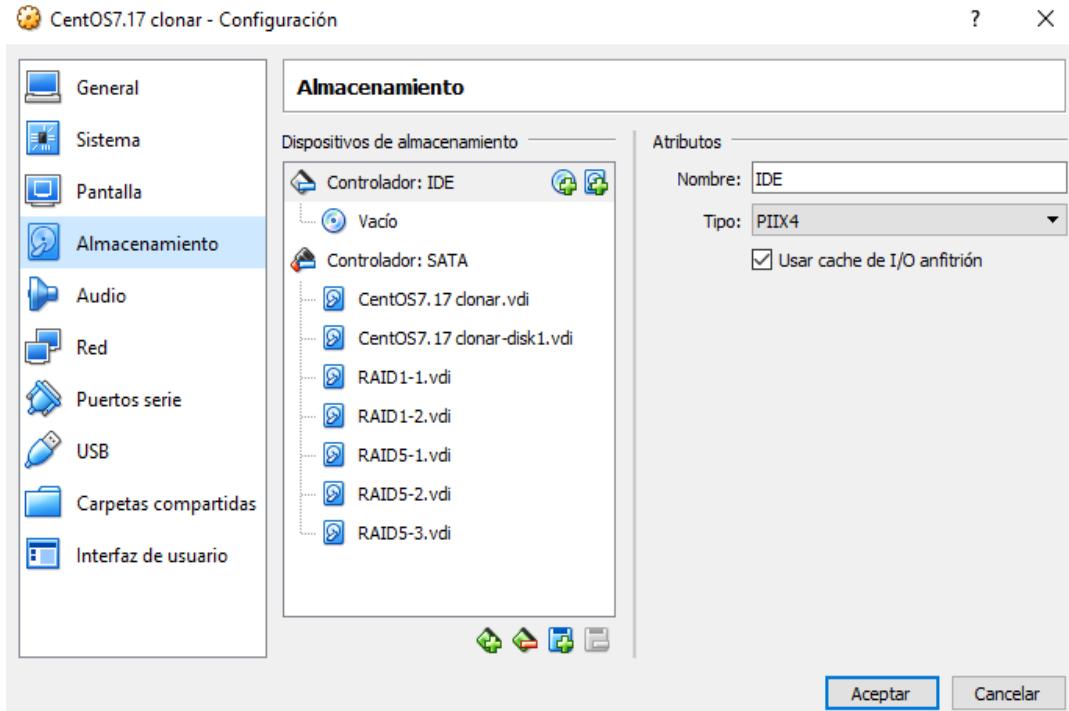
```

programador@debian9: ~
programador@debian9: ~
root@debian9:/home/programador/backupServidor# ls
backup
root@debian9:/home/programador/backupServidor# cd backup/
root@debian9:/home/programador/backupServidor/backup# ls
copia_seguridad
root@debian9:/home/programador/backupServidor/backup# cd copia_seguridad/
root@debian9:/home/programador/backupServidor/backup/copia_seguridad# ls
root@debian9:/home/programador/backupServidor/backup/copia_seguridad#

```

## Raid

Para raid he clonado mi máquina, por si podía a ver jodido algo. Para prepararlo vamos a hacer lo siguiente:



- Clonamos la máquina y después añadimos dándole click derecho al controlador SATA:
  - Añadir disco, y asignamos **2 para RAID 1** y **3 para RAID 5**.
- Creamos el raid dentro de la máquina de la siguiente forma:
  - Para **RAID 1**:

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdc /dev/sdd
```

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sde /dev/sdf /dev/sdg
```

- Desde dentro de la máquina ponemos **cat /proc/mdstat**. Con esto podemos ver los discos de raid 1 y raid 5 correspondientes:

```
Programador@programador:/home/Programador
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador Programador]# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md1 : active raid5 sdg[3] sdf[1] sde[4]
      16760832 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

md0 : active raid1 sdd[1] sdc[2]
      8372224 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

- Provocamos un fallo en uno de los discos:

```
[root@programador Programador]# mdadm /dev/md0 --fail /dev/sdc --remove /dev/sdc
mdadm: set /dev/sdc faulty in /dev/md0
mdadm: hot removed /dev/sdc from /dev/md0
[root@programador Programador]#
```

- Añadimos el disco de nuevo y comprobamos como está el disco:

```
[root@programador Programador]# mdadm /dev/md0 --add /dev/sdc
mdadm: added /dev/sdc
[root@programador Programador]# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md1 : active raid5 sdg[3] sdf[1] sde[4]
      16760832 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

md0 : active raid1 sdc[2] sdd[1]
      8372224 blocks super 1.2 [2/1] [_U]
      [==>.....] recovery = 16.5% (1389184/8372224) finish=0.5min speed=2
31530K/sec

unused devices: <none>
```

Como podemos observar el disco de RAID 1 se está recuperando después de haber provocado el fallo.

# Freebsd

## Antes de comenzar

Como en CentOS deshabilitamos el selinux y el firewalld y trabajamos con todo eso desactivado para evitar posibles problemas en el futuro.

## Servidor Web (Dominios virtuales)

- Instalaremos apache con el comando:
  - **pkg install apache22**
- Nos dirigiremos al directorio **/usr/local/etc/apache24/extra** y configuraremos el archivo **httpd-vhosts.conf**

Añadiremos esto por cada dominio que queramos añadir al servidor web

```
<VirtualHost *:80>
    DocumentRoot "/usr/local/www/apache24/FreeBSD"
    ServerName FreeBSD.com
    ServerAlias www.FreeBSD.com
    <Directory "/usr/local/www/apache24/FreeBSD">
        Allow from all
    </Directory>
</VirtualHost>
```

Crearemos la carpeta con el directorio que pusimos antes con el nombre del domino.

```
cgi-bin data error icons
root@FreeBSD:/usr/local/www/apache24 # mkdir FreeBSD
root@FreeBSD:/usr/local/www/apache24 # chmod 777 FreeBSD/
root@FreeBSD:/usr/local/www/apache24 #
```

- Crearemos un index.HTML dentro de la carpeta que será lo que mostrará el servidor web cuando accedamos al dominio.
- Para la instalación de los **CMS** hemos optado por WordPress y Joomla.

Instalar previamente las extensiones de PHP:

```
pkg install php56-mysqli php56-curl php56-zlib
```

Una vez descarguemos el .tar deberemos mover el **CMS** descargado a la carpeta del dominio, mismo proceso para los dos:

```
cp -r wordpress/* /usr/local/www/apache24/data/
```

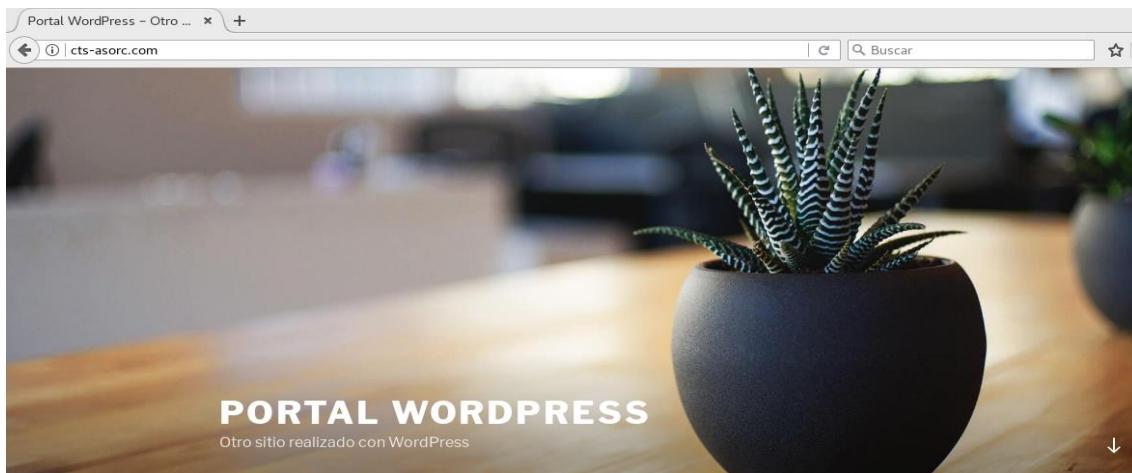
Por último, crearemos el nombre de la base de datos , usuario y contraseña:

```
define('DB_NAME', 'wordpress');
...
define('DB_USER', 'wpuser');
...
define('DB_PASSWORD', 'wppassword');
```

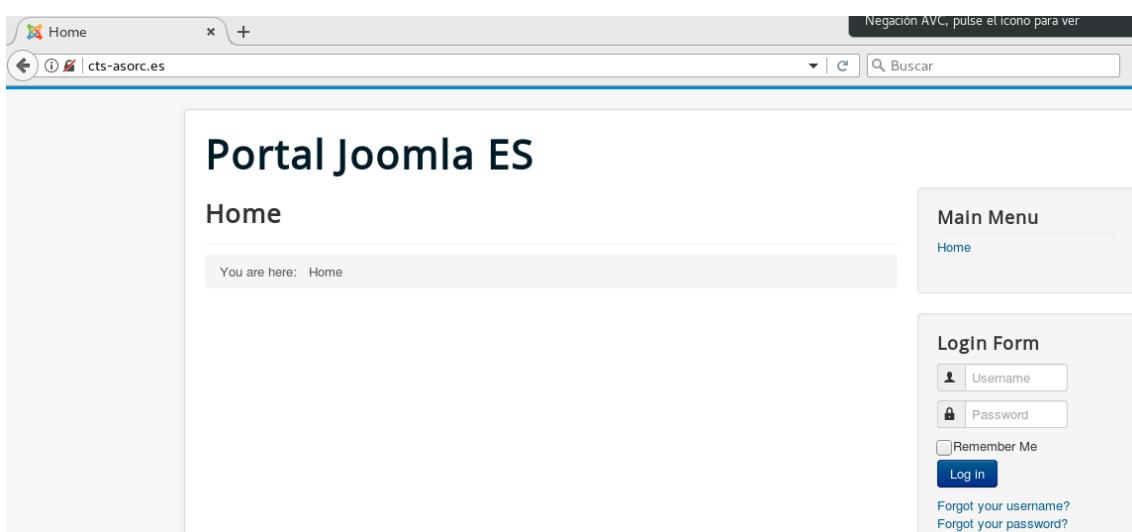
- Reiniciamos el servicio web:
  - **service apache22 enable && service apache22 start**

Y su posterior comprobación del servicio desde el cliente Linux(comprobar el resolv.conf por posibles fallos):

- Para el dominio web **cts-asorc.com**:



- Para el dominio web **cts-asorc.es**:



# Servidor Base de datos (MySQL)

- Nos dirigiremos a los puertos para descargar MySQL
  - nano /usr/ports/databases/mysql56-server
  - Make install clean

```
root@FreeBSD:/usr/ports/databases/mysql56-server # make install clean
```

- Despues de esperar mucho tiempo de instalación, cambiamos el propietario y grupo del directorio de datos:
  - chown -R mysql/var/db/mysql/

Añadiremos a **/etc/rc.conf** **mysql\_enable="YES"**

A continuación, una vez arrancado el servicio, cambio la contraseña del usuario root, que es el que voy a utilizar para acceder al servidor de base de datos --> Mysqladmin -u root password 123

Para acceder a la base de datos utilizaremos el comando mysql -u root -p

- Podremos crear una base de datos:
  - CREATE DATABASE prueba;

*Use prueba;*

*CREATE TABLE Pruebas{*

*Id varchar(8),*

*Nombre varchar(6)*

*};*

- Una vez instalado el php de mysql creamos el archivo de configuración de php, para crearlo copiamos uno que ya hay por defecto y le cambiamos el nombre por php.ini y añadirnos el **index.php**

```
<IfModule dir_module>
    DirectoryIndex index.html index.php index.htm
```

- Una vez hecho esto añadiremos el .php en /usr/local/www/apache24/data y tan solo quedaría configurar el php como queramos.

Comprobaremos el servicio en el cliente Linux, desde de dentro de la carpeta que lo contenga, en nuestro caso será **cts-asorc.com/db.php**

<b>id</b>	<b>nombre</b>	<b>edad</b>
1	Javi	20
2	Francisco	14
3	Oscar	26
4	Angel	20
5	Ronald	21

Y como podemos observar funciona correctamente.

## Instant Messaging (Jabber)

Tiene el mismo procedimiento que CentOS.

- Primero instalamos el paquete correspondiente:
  - **/usr/ports/net-im/openfire --> make install clean**
- Iniciamos el servicio correspondiente:
  - **Systemctl start openfire && systemctl enable openfire**

Nos vamos a Firefox, y ponemos **192.168.56.101:9090** y nos saldrá los siguientes pasos:



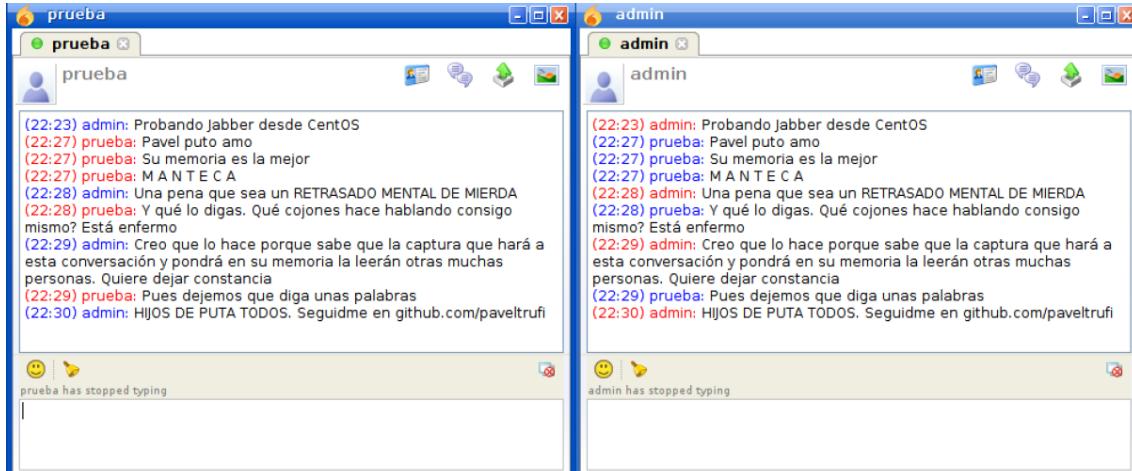
- Elegimos el nombre del usuario a crear, en nuestro caso será programador.

## Crear Usuario

Use el formulario siguiente para crear un nuevo usuario.

Crear Nuevo Usuario	
Usuario: *	<input type="text" value="programador"/>
Nombre:	<input type="text" value="programador"/>
Email:	<input type="text" value="programador@jabber.com"/>
Contraseña: *	<input type="password" value="*****"/>
Confirmar Contraseña: *	<input type="password" value="*****"/>
Is Administrator?	<input type="checkbox"/> (Grants admin access to Openfire)
<input type="button" value="Crear Usuario"/> <input type="button" value="Crear y Crear Otro"/> <input type="button" value="Cancelar"/>	

- Desde una máquina cliente como es Linux y desde nuestro Freebsd nos conectamos al Spark:



Y comprobamos que funciona correctamente.

## Servidor Proxy (Squid)

- Primero instalaremos el paquete correspondiente, para ello nos dirigiremos a los puertos para realizar la descarga.
  - `cd /usr/ports/www/squid`
  - `make install clean`
- Añadir en `/etc/rc.conf` la línea `squid_enable = YES`

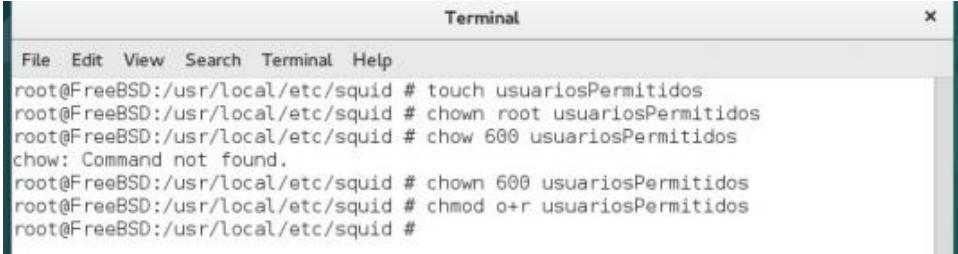
Para configurar el servicio no dirigiremos a `squid.conf`:

- Denegaremos el acceso a páginas Web como en nuestro caso es el [www.marca.com](http://www.marca.com), contenido de imágenes y autentificación por usuario.

Añadiremos estas dos líneas en las reglas para bloquear el contenido de las imágenes y bloquear el acceso a la página web www.marca.com.

```
acl denegado url_regex "/usr/local/etc/squid/paginasDenegadas"  
acl http_access deny denegado
```

- Para los usuarios crearemos un archivo donde se almacenarán los usuarios que creamos y sus contraseñas cifradas.



The screenshot shows a terminal window titled "Terminal". The command history at the top includes:

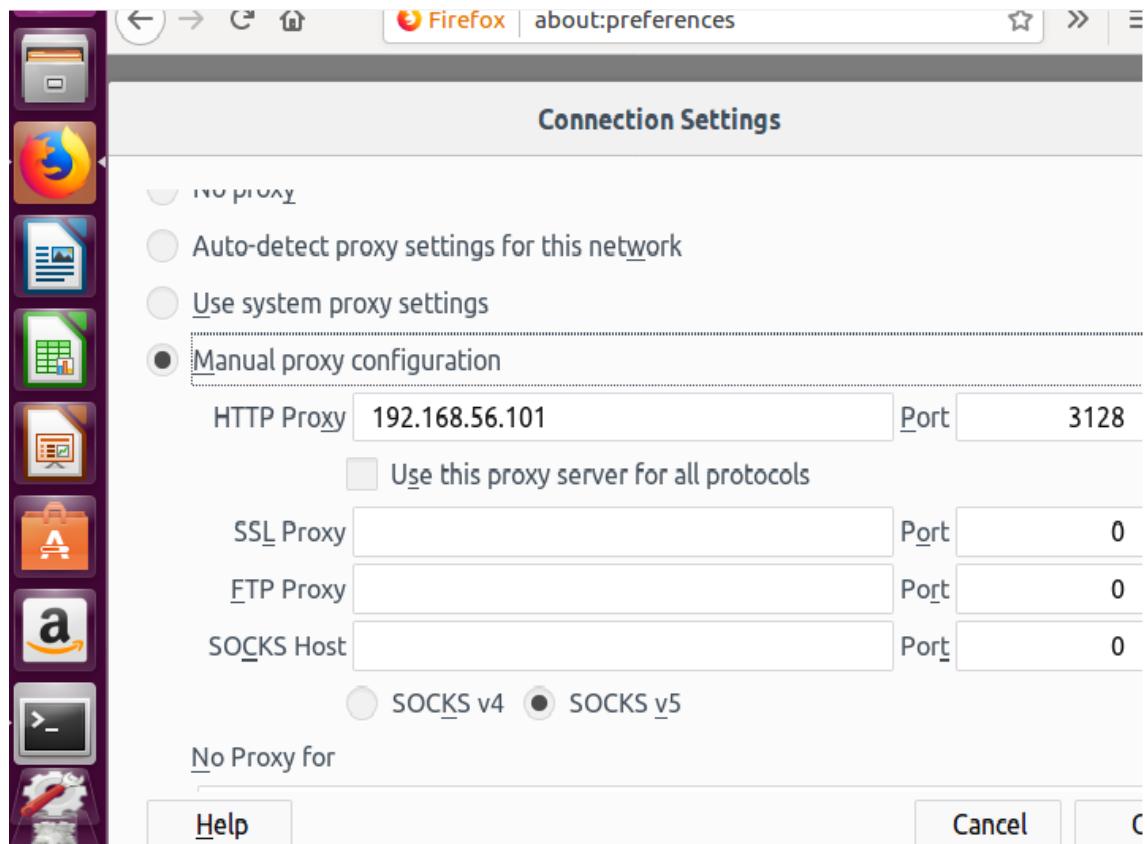
```
File Edit View Search Terminal Help  
root@FreeBSD:/usr/local/etc/squid # touch usuariosPermitidos  
root@FreeBSD:/usr/local/etc/squid # chown root usuariosPermitidos  
root@FreeBSD:/usr/local/etc/squid # chow 600 usuariosPermitidos  
chow: Command not found.  
root@FreeBSD:/usr/local/etc/squid # chown 600 usuariosPermitidos  
root@FreeBSD:/usr/local/etc/squid # chmod o+r usuariosPermitidos  
root@FreeBSD:/usr/local/etc/squid #
```

The user then enters the following configuration lines:

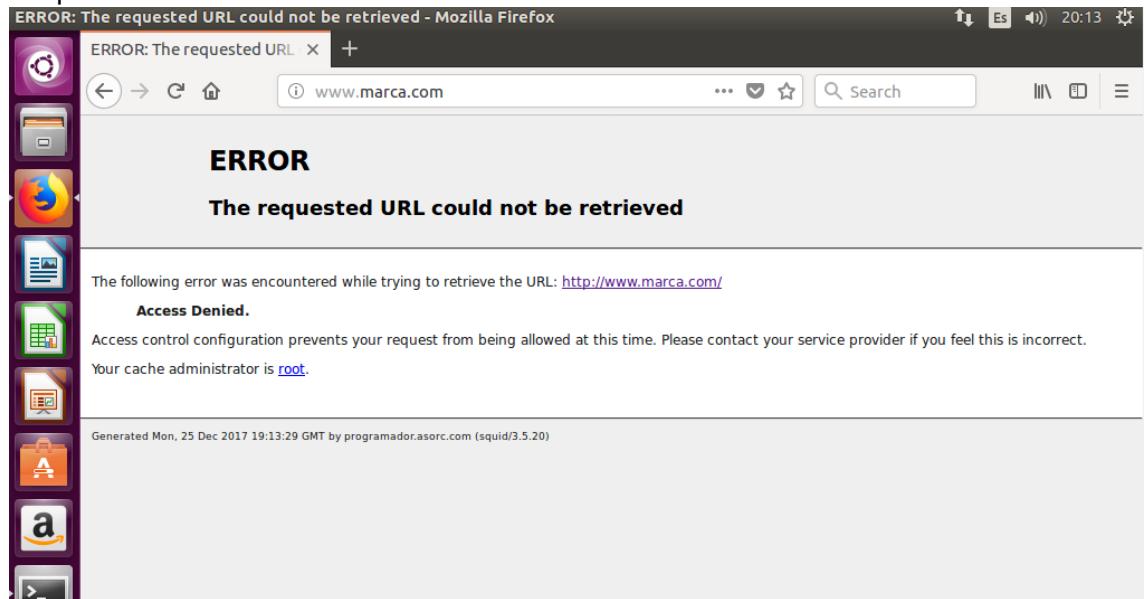
```
auth_param basic program /usr/local/libexec/squid/basic_ncsa_auth  
/usr/local/etc/squid/usuariosPermitidos  
acl password proxy_auth REQUIRED
```

- Iniciamos el sistema y si hace falta lo reiniciamos:
  - `Systemctl enable squid && systemctl start squid`

Nos vamos a la máquina cliente en nuestro caso Linux y configuramos el proxy en el Firefox → Preferencias → Proxy settings.



Y comprobamos que el servicio funciona correctamente para la página [www.marca.com](http://www.marca.com) que la bloquea exitosamente:



Y por ejemplo [www.facebook.com](http://www.facebook.com) no la bloquea:

ERROR: The requested URL [Facebook - Log In or Sign Up](#)

To help personalize content, tailor and measure ads, and provide a safer experience, we use cookies. By clicking or navigating the site, you agree to allow our collection of information on and off Facebook through cookies. Learn more, including about available controls: [Cookies Policy](#).

**facebook**

Email or Phone      Password

Forgot account?

**Sign Up**

It's free and always will be.

First name      Last name

Mobile number or email

New password

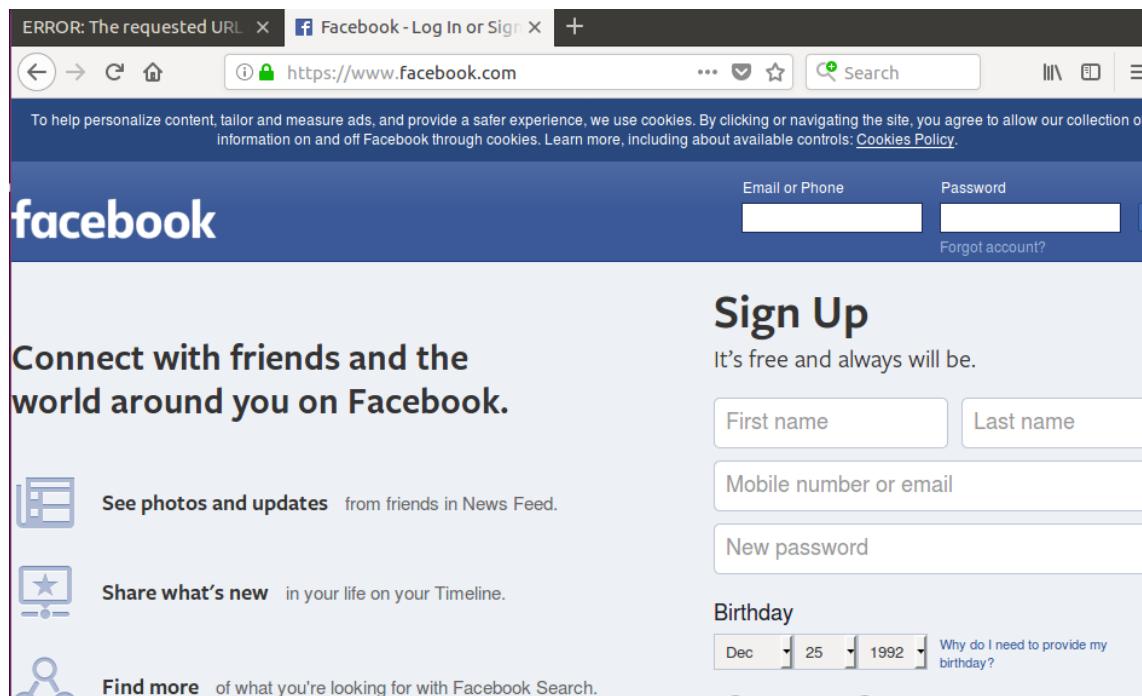
Birthday

Dec 25 1992 [Why do I need to provide my birthday?](#)

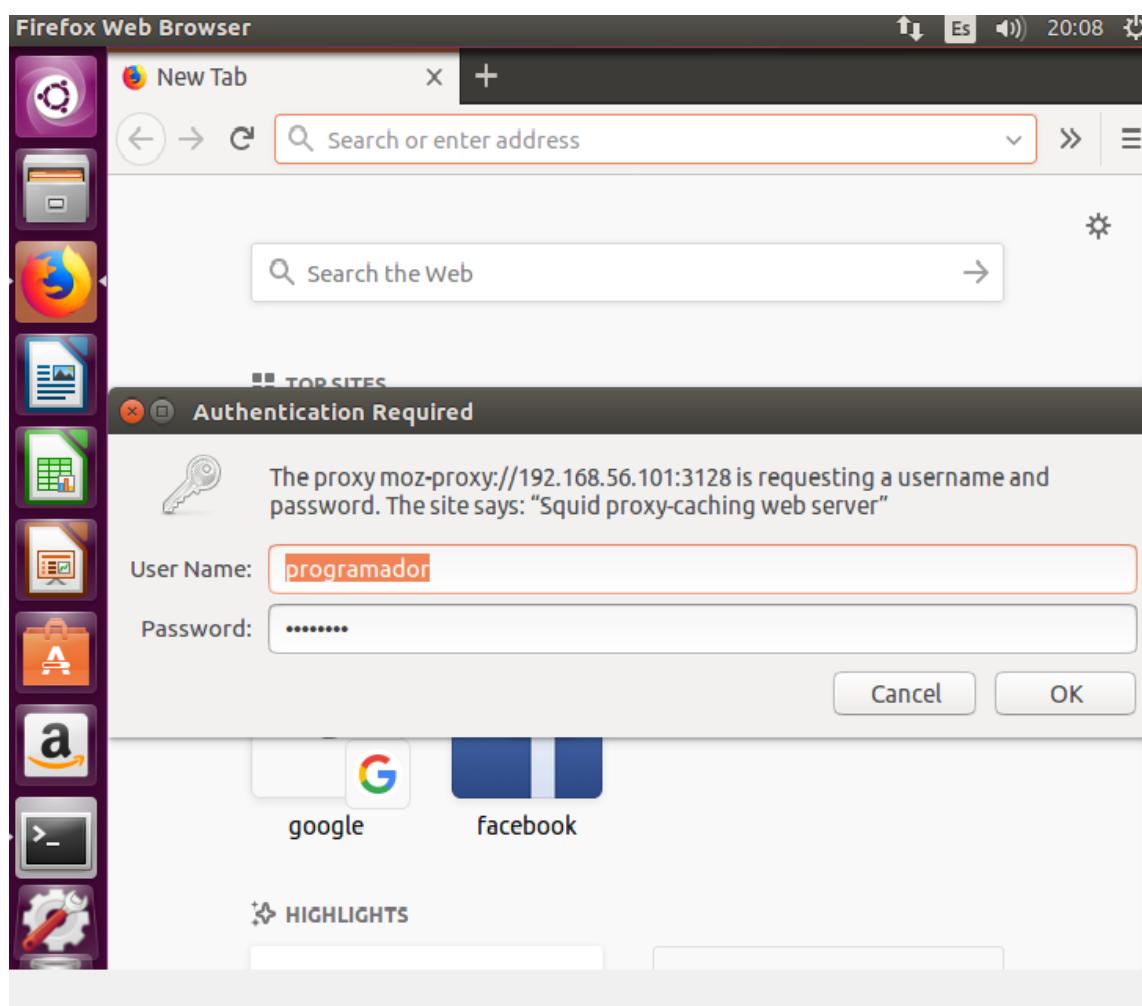
See photos and updates from friends in News Feed.

Share what's new in your life on your Timeline.

Find more of what you're looking for with Facebook Search.

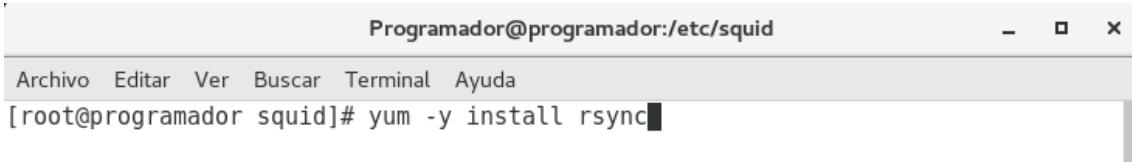


Cada vez que inicie el Firefox pedirá **autentificación por usuario** como se muestra a continuación:



# Backup (Sincronización, absolutos, incremental)

- El paquete ya se encuentra instalado, pero por si acaso he utilizado **rsync**.



```
Programador@programador:/etc/squid
-
Archivo Editar Ver Buscar Terminal Ayuda
[root@programador squid]# yum -y install rsync
```

Comento, este servicio desde mi punto de vista se puede hacer tanto en servidor, como en cliente, pero deberemos de enseñar que funcionar y comprendemos correctamente el servicio. Para ello deberemos de mostrar los siguientes pasos:

```
root@debian9:/home/programador/Escritorio# cat backup
1 - Desde el cliente nos conectamos al servidor:
rsync -avz -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor

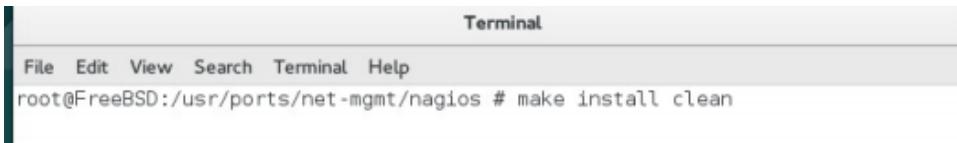
2 - creamos dentro de la carpeta del servidor -> /home/Programador/backup/copia
-> COMPLETO
cd /home/Programador/backup/copia -> touch prueba.txt

3 - nos vamos al cliente y ejecutamos:
INCREMENTAL-> rsync -avr -e ssh Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
4 - eliminamos el prueba.txt ----> SINCRONIZACIÓN
rsync -rtvy --delete-delay Programador@192.168.56.101:/home/Programador/backup /home/programador/backupServidor
root@debian9:/home/programador/Escritorio#
```

Este servicio es igual para todos los sistemas operativos (echar un vistazo a Backup de CentOS).

## Monitorización de servicios (Nagios)

- Instalaremos nagios mediante los puertos :



```
Terminal
File Edit View Search Terminal Help
root@FreeBSD:/usr/ports/net-mgmt/nagios # make install clean
```

- Añadiremos **nagios\_enable = YES**

Para configurar nagios usaremos los ejemplos que nos da nagios quitando la extensión ejemplo:

```

root@FreeBSD:~ # cd /usr/local/etc/nagios/
root@FreeBSD:/usr/local/etc/nagios # ls
cgi.cfg-sample      objects
nagios.cfg-sample   resource.cfg-sample
root@FreeBSD:/usr/local/etc/nagios # cp cgi.cfg-sample cgi.cfg
root@FreeBSD:/usr/local/etc/nagios # cp nagios.cfg-sample nagios.cfg
root@FreeBSD:/usr/local/etc/nagios # cp resource.cfg-sample resource.cfg
root@FreeBSD:/usr/local/etc/nagios # cd objects/
root@FreeBSD:/usr/local/etc/nagios/objects # ls
commands.cfg-sample   printer.cfg-sample   timeperiods.cfg-sample
contacts.cfg-sample   switch.cfg-sample
localhost.cfg-sample  templates.cfg-sample
root@FreeBSD:/usr/local/etc/nagios/objects # cp commands.cfg-sample commands.cfg
root@FreeBSD:/usr/local/etc/nagios/objects # cp printer.cfg-sample printer.cfg
root@FreeBSD:/usr/local/etc/nagios/objects # cp timeperiods.cfg-sample timeperiods.cfg
root@FreeBSD:/usr/local/etc/nagios/objects # cp contacts.cfg-sample contacts.cfg
root@FreeBSD:/usr/local/etc/nagios/objects # cp switch.cfg-sample switch.cfg
root@FreeBSD:/usr/local/etc/nagios/objects # cp localhost.cfg-sample localhost.cfg
root@FreeBSD:/usr/local/etc/nagios/objects # cp templates.cfg-sample templates.cfg
root@FreeBSD:/usr/local/etc/nagios/objects #

```

Copiaremos todos estos archivos en directorio raíz del servidor web , para que cuando pongamos al ip/nagios podremos acceder:

```

root@FreeBSD:/usr/local/etc/nagios/objects # mkdir /usr/local/www/apache24/data/nagios
root@FreeBSD:/usr/local/etc/nagios/objects # ln -s /usr/local/www/apache24/data/nagios/
root@FreeBSD:/usr/local/etc/nagios/objects # ln -s /usr/local/www/nagios/* /usr/local/www/apache24/data/nagios/

```

- Importante descomentar la siguiente línea en **httpd.conf**:

```

<IfModule !mpm_prefork_module>
    #LoadModule cgid_module libexec/apache24/mod_cgid.so
</IfModule>
<IfModule mpm_prefork_module>
    LoadModule cgi_module libexec/apache24/mod_cgi.so

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

```

- Para su comprobación el profesor solamente pedirá en el cliente Linux que se muestra la página de **Nagios** correctamente:

The screenshot shows the Nagios Core interface in Mozilla Firefox. The main title bar says "Nagios Core - Mozilla Firefox". The address bar shows "192.168.7.130/nagios/". The left sidebar has sections for General, Current Status (with sub-options like Host Groups, Services, and Hosts), and a search bar. The main content area displays "Current Network Status" with last updated information and host/service counts. It also shows "Host Status Totals" and "Service Status Totals" with counts for Up, Down, Unreachable, Pending, Ok, Warning, Unknown, Critical, and Pending states. Below this is a table titled "Service Status Details For All Hosts" with columns for Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The table lists various services like Current Load, Current Users, HTTP, PING, SSH, Swap Usage, and Total Processes, along with their status and specific metrics.

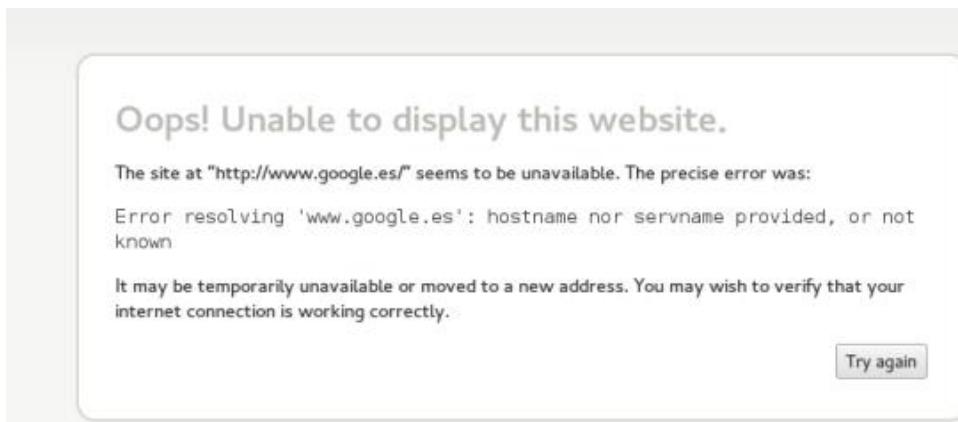
## Firewall

No hace falta instalar ningún paquete porque ya viene instalado predeterminado, por lo que tendremos que modificar un archivo de configuración:

- Nos vamos a `/etc/pf.conf` y añadiremos `pf_enable="YES"`
- Dentro del archivo de configuración `/etc/pf.conf` añadiremos las reglas que queramos.

Lo único que tendremos que hacer iniciar el servicio(no habilitarlo con enable) porque esté bloqueará los demás servicios.

- `Systemctl start pf`



## Windows Server 2016

Antes de comenzar

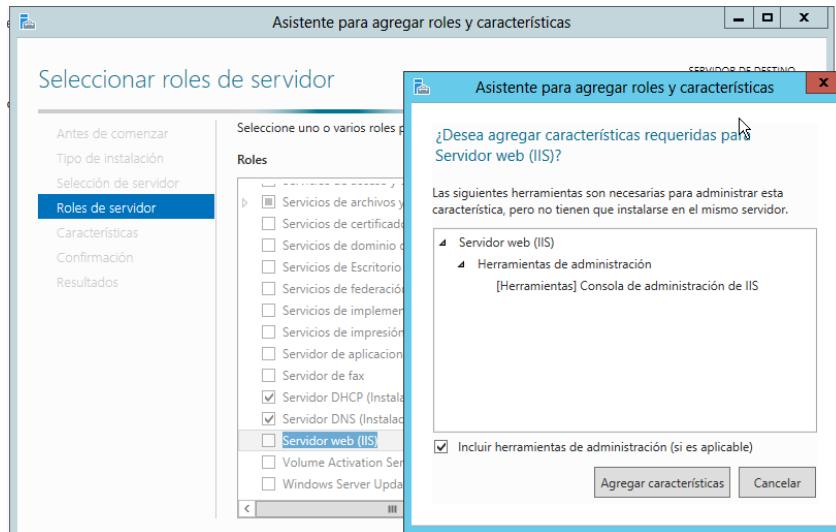
- Desactivaremos el firewall y el selinux.

## Servidor Web (Dominios virtuales)

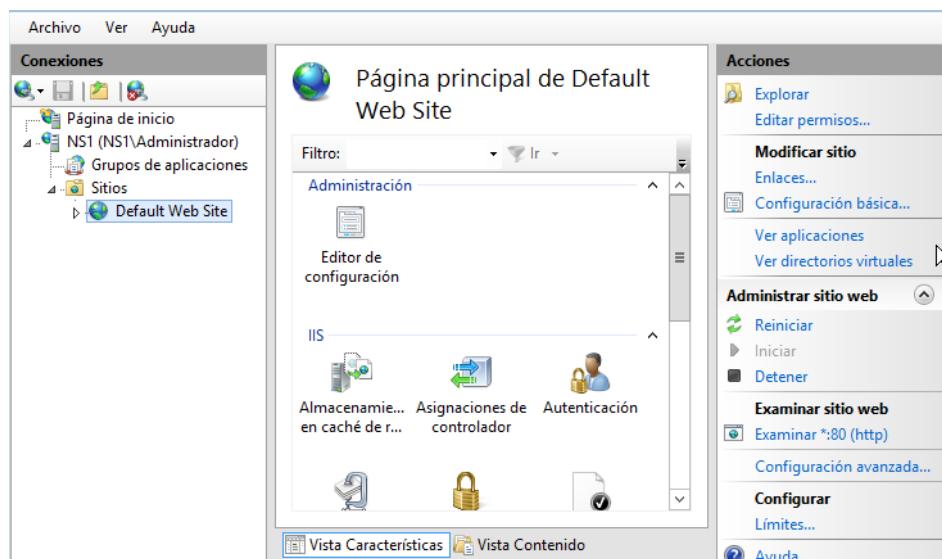
- En cuanto a la configuración del servicio de servidor Web, entramos al administrador del servidor y procedemos a instalar los roles y características del IIS, que nos permitirá configurar el servicio:



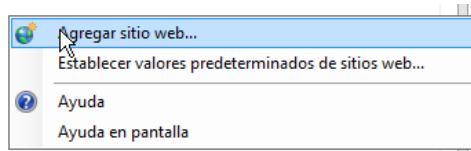
- Tras elegir las características del servidor, agregamos el rol Servidor Web IIS:



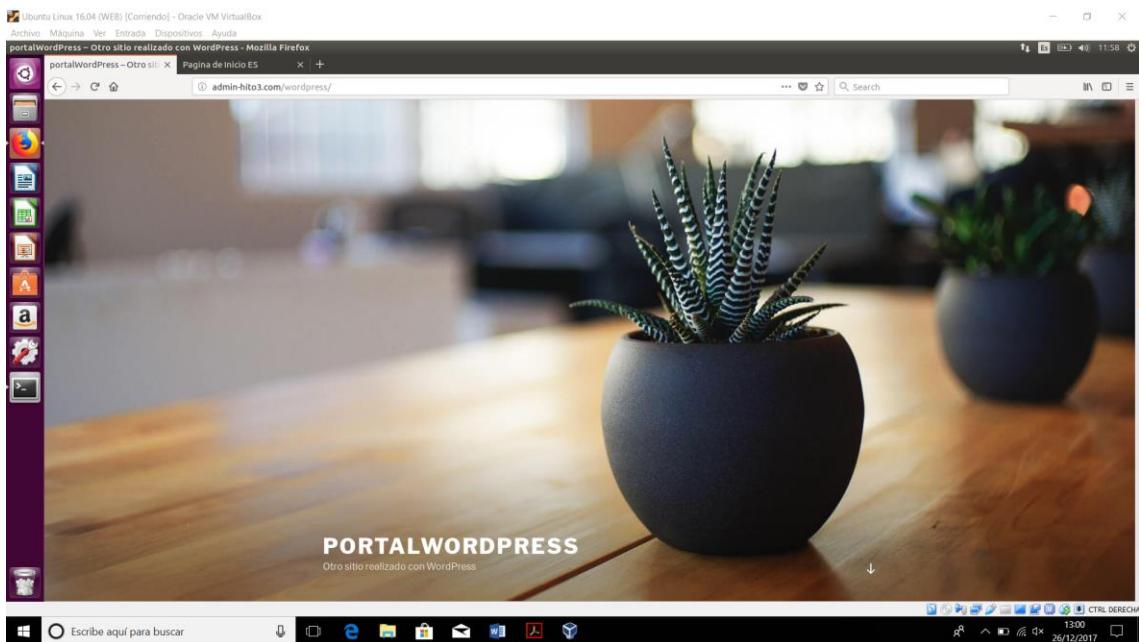
- Una vez agregado todo el rol, debemos añadir la herramienta de CGI, y pasar a configurar los sitios web y las páginas principales:



- Pinchando sobre sitios y botón derecho, en el desplegable seleccionamos **Agregar sitio web...** :

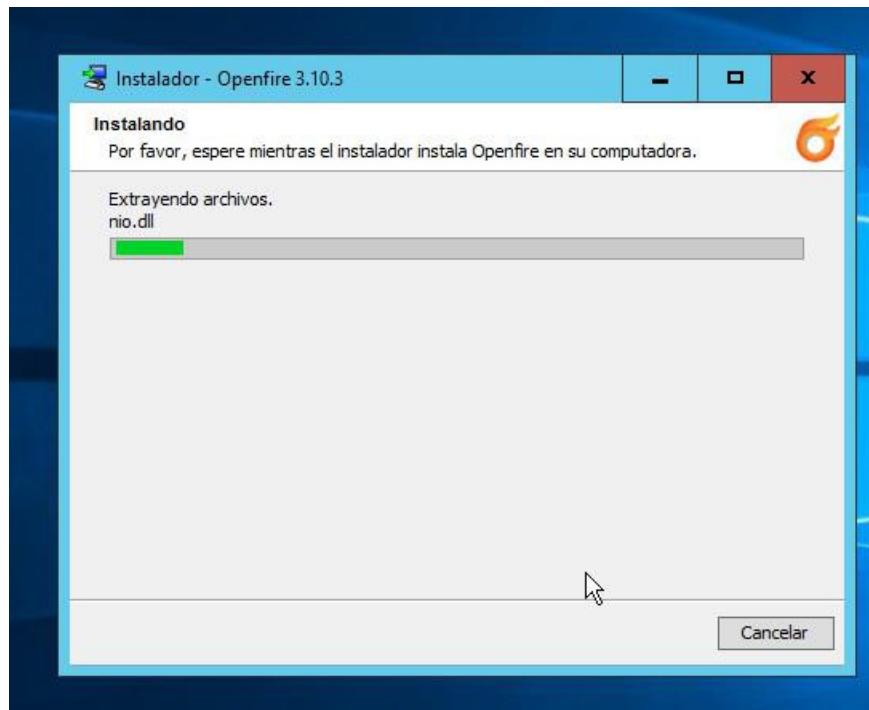


- Donde nos aparecerá una ventana de configuración y en la cual, después de haber configurado correctamente nuestros dos servicios de nombres, seleccionaremos las características elegidas y aceptaremos.
- Y continuamos para asignar los CMS a nuestros dominios virtuales, para lo que he elegido Windows platform web, y configurando Joomla y wordpress para los dominios de **admin-hito3.com** y **admin-hito3.es** respectivamente, obteniendo como resultado al consultar las páginas en nuestro navegador:



## Instant Messaging (Jabber)

- En primer lugar, para configurar el servicio de mensajería instantánea, instalo y configuro Openfire:



- Una vez instalado, basta con seguir los pasos indicados, en los que he decidido dejar los puertos predefinidos para este servicio, por tanto, el puerto empleado será el 9090 y accediendo a nuestra IP 192.168.56.103:9090 configuraremos el administrador del servicio:

A screenshot of a web-based setup interface for Openfire. At the top, there is a browser-style header with back, forward, and refresh buttons, and a URL bar containing "localhost:9090/setup/index.jsp". Below the header is the Openfire logo, which is a stylized orange flame icon followed by the word "openfire". The main interface has a tab navigation bar with "Setup" selected. On the left side, there is a sidebar menu with the following items: "Setup Progress" (which shows a progress bar), "Language Selection" (which is currently active and highlighted in yellow), "Server Settings", "Database Settings", "Profile Settings", and "Admin Account". The main content area is titled "Welcome to Setup" and contains the message "Welcome to Openfire Setup. This tool will lead you through the i preferred language." Below this, there is a section titled "Choose Language" with four radio buttons:

- Czech (cs\_CZ)
- Deutsch (de)
- English (en)
- Español (es)

Puerto de la Consola de Administración:  [?](#)  
Puerto de la Consola de Administración Segura:  [?](#)  
Property Encryption via: [?](#)  
 Blowfish  
 AES  
Property Encryption Key:

- Configurado lo cual, pasamos a indicar la información de quien administrará la cuenta del servicio, para lo que he empleado una cuenta [admin@example.com](mailto:admin@example.com):

### Cuenta del Administrador

Ingrese la configuración para la cuenta del administrador del sistema (nombre de usuario "admin"). Es importante una contraseña que no pueda ser adivinada fácilmente, por ejemplo que tenga al menos seis caracteres y una mezcla de mayúsculas, minúsculas y números. Puede saltar este paso si ya ha configurado su cuenta de administrador (no recomendado para usuarios principiantes).

Email del Administrador:   
Una dirección de email válida para la cuenta del administrador.  
Nueva Contraseña:  [I](#)  
Confirme la Contraseña:

- Y para poder comprobar el correcto funcionamiento del servicio configurado, seguimos diferentes pasos, creamos una cuenta para otro usuario, en mi caso programador:

**Crear Usuario**

Use el formulario siguiente para crear un nuevo usuario.

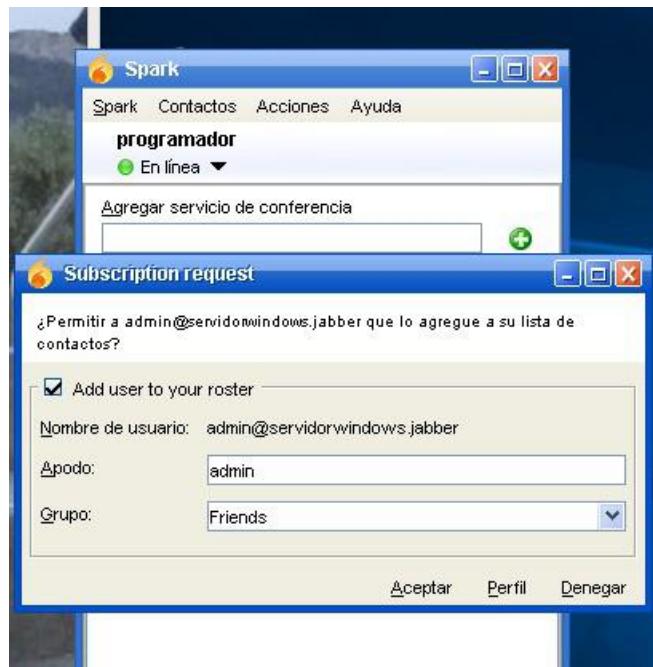
**Crear Nuevo Usuario**

Usuario: *	<input type="text" value="programador"/>
Nombre:	<input type="text" value="programador"/>
Email:	<input type="text" value="programador@jabber.com"/>
Contraseña: *	<input type="password"/>
Confirmar Contraseña: *	<input type="password"/>
Is Administrator?	<input type="checkbox"/> (Grants admin access to Openfire)
<input type="button" value="Crear Usuario"/> <input type="button" value="Crear y Crear Otro"/> <input type="button" value="Cancelar"/>	

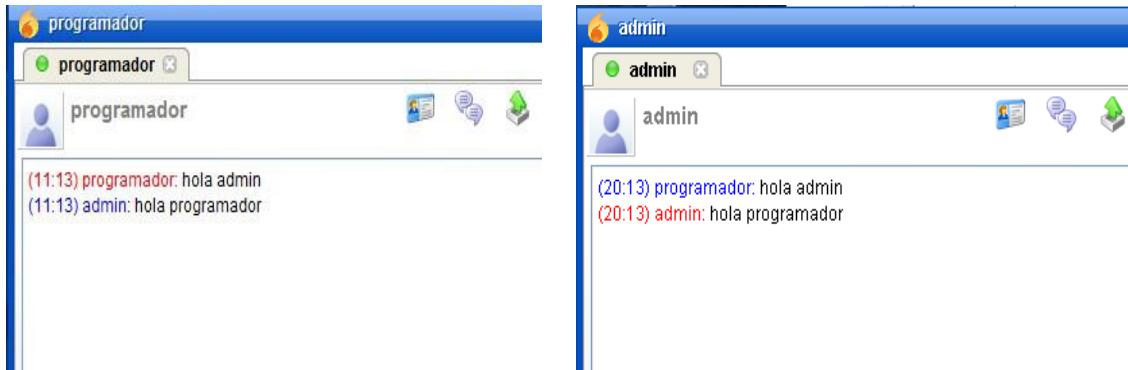
\* Campos Requeridos

- Y realizado el primer paso, indicado justo arriba, pasamos a descargar e instalar el cliente de mensajería instantánea SPARK, donde accediendo con nuestras dos cuentas

desde dos clientes diferentes, añadimos el contacto de programador a la lista de amigos y viceversa con la cuenta de **admin**:

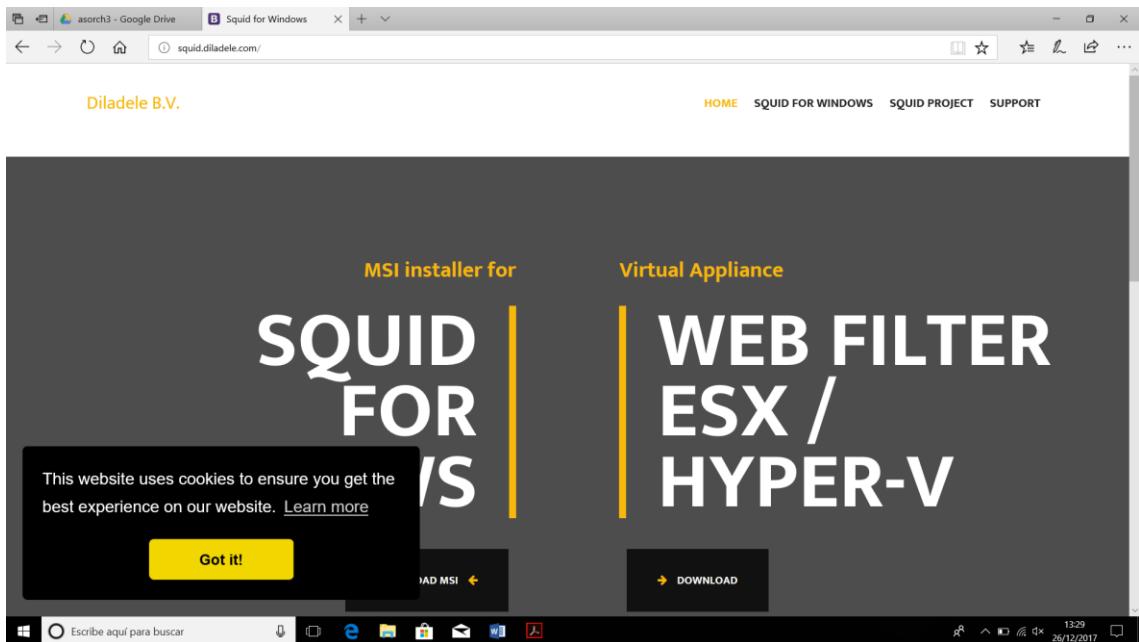


- Y tras ello, podemos comprobar la correcta comunicación entre ambos clientes como se puede observar a continuación:



## Servidor Proxy (Squid)

- Para la configuración del proxy de nuestro sistema he elegido configurar Squid, para lo cual buscamos y lo descargamos:



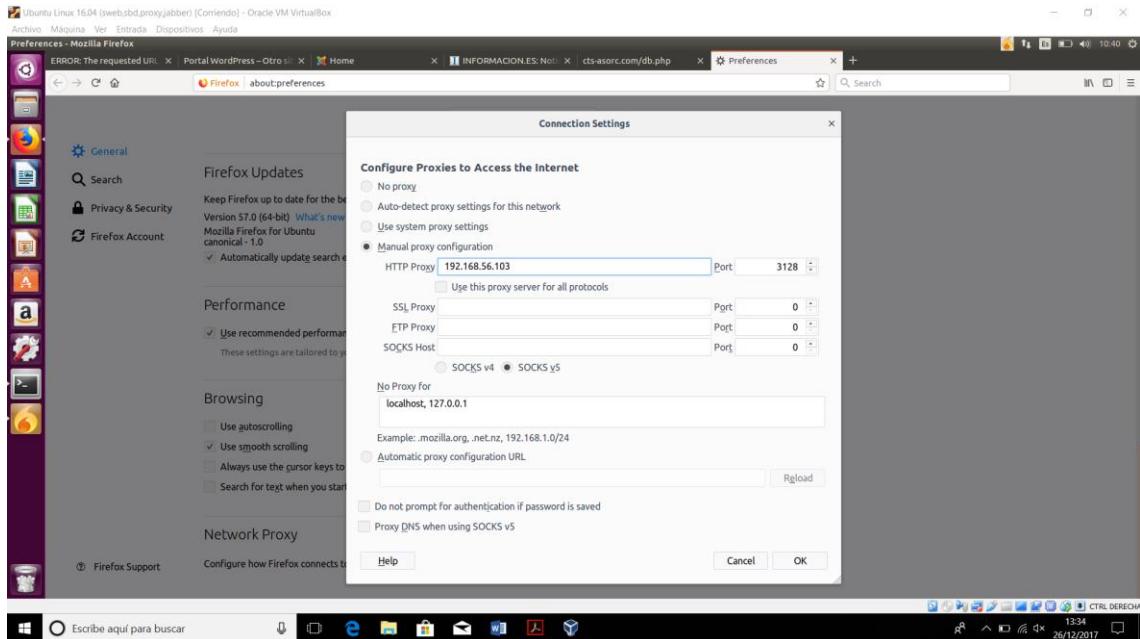
- Añadimos 3 reglas, una para que opere sobre la red el proxy, otra para denegar webs y otra para denegar contenidos y activamos esas reglas, para que las utilice.

```

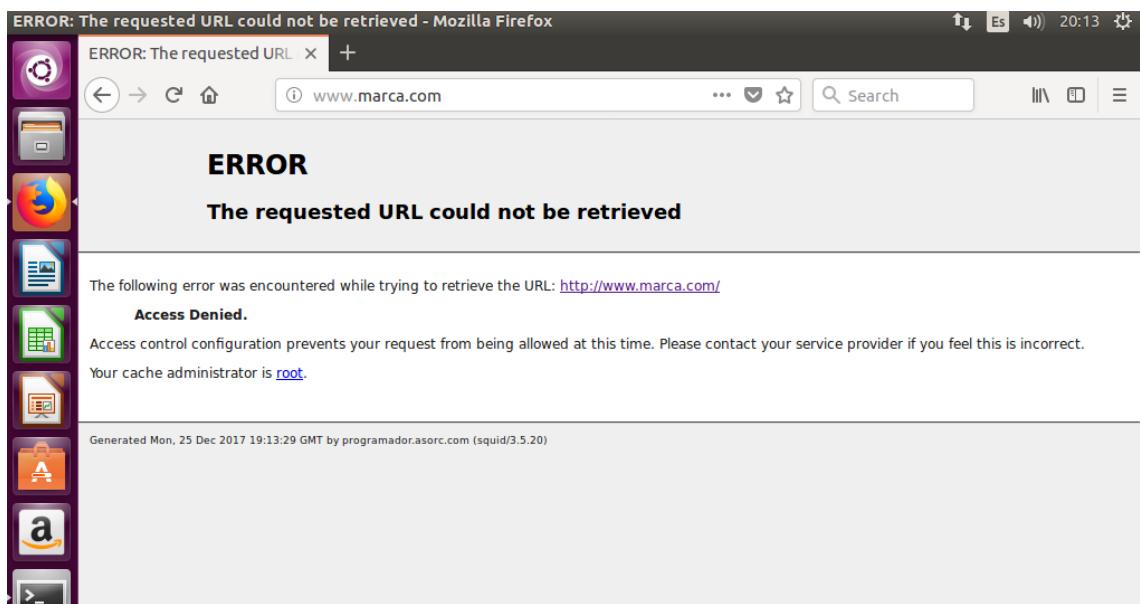
squid: Bloc de notas
Archivo Edición Formato Ver Ayuda
#       # attribute is one of DN/C/O/CN/L/ST
#
#       acl aclname ext_user username ...
#       acl aclname ext_user_regex [-i] pattern ...
#           # string match on username returned by external acl helper
#           # use REQUIRED to accept any non-null user name.
#
#Examples:
#acl macaddress arp 09:00:2b:23:45:67
#acl myexample dst_as 1241
#acl password proxy_auth REQUIRED
#acl fileupload req_mime_type -i ^multipart/form-data$
#acl javascript rep_mime_type -i ^application/x-javascript$
#
#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl ws src 192.168.56.0/24
acl denywebs url_regex "C:\squid\denywebs.txt"
acl denycontents url_regex "C:\squid\denycontents|.txt"
#
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8      # RFC1918 possible internal network
acl localnet src 172.16.0.0/12    # RFC1918 possible internal network
acl localnet src 192.168.0.0/16   # RFC1918 possible internal network
#
acl SSL_ports port 443          # http
acl Safe_ports port 80            # ftp
acl Safe_ports port 21            # https
acl Safe_ports port 443           # gopher
acl Safe_ports port 70            # wais
acl Safe_ports port 210           # 

```

- Mediante el icono de escritorio, iniciamos el servicio, y configuramos el navegador para que emplee nuestro servicio proxy:

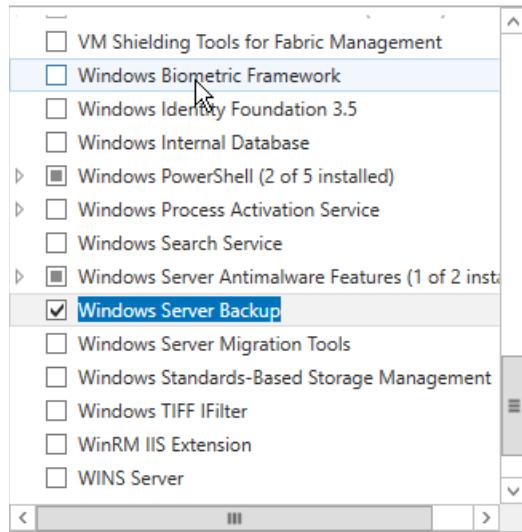


- Y comprobamos el correcto funcionamiento del proxy, comprobando el bloqueo de la página bloqueada marca.com y la no aparición de imágenes en una web como diarioinformacion.com, como se puede observar a continuación:

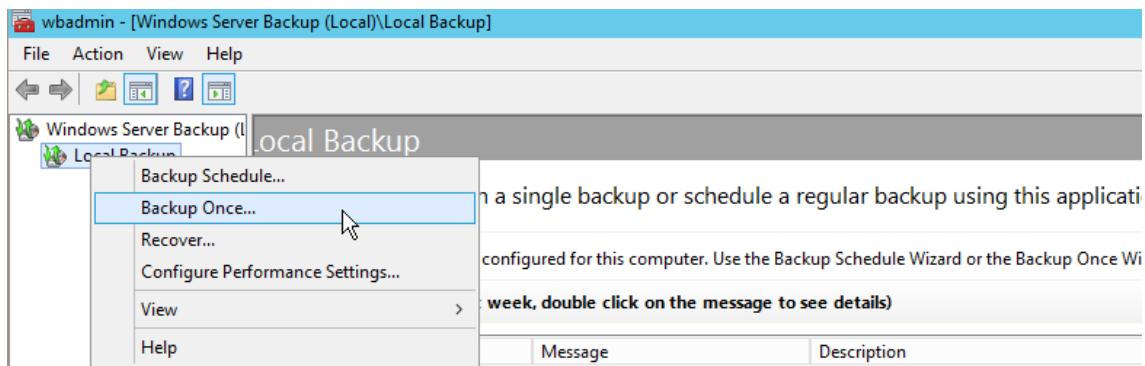


## Backup (Copias de seguridad)

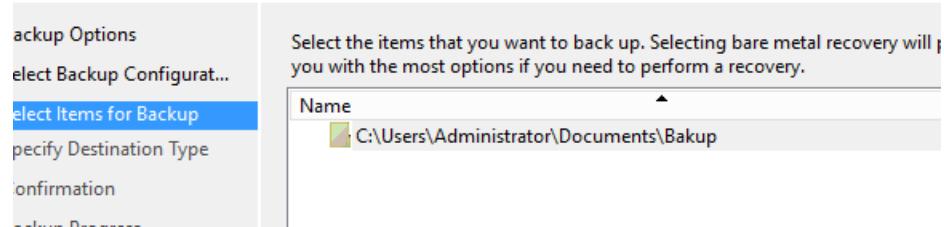
- En este caso, al igual que en servidor web, accedemos a roles del servidor, donde agregamos el servidor de Copias de Seguridad de Windows:



- Una vez instalado y configurado, entrando en herramientas del administrador y seleccionando el rol recientemente añadido, pasamos a realizar una copia de seguridad:



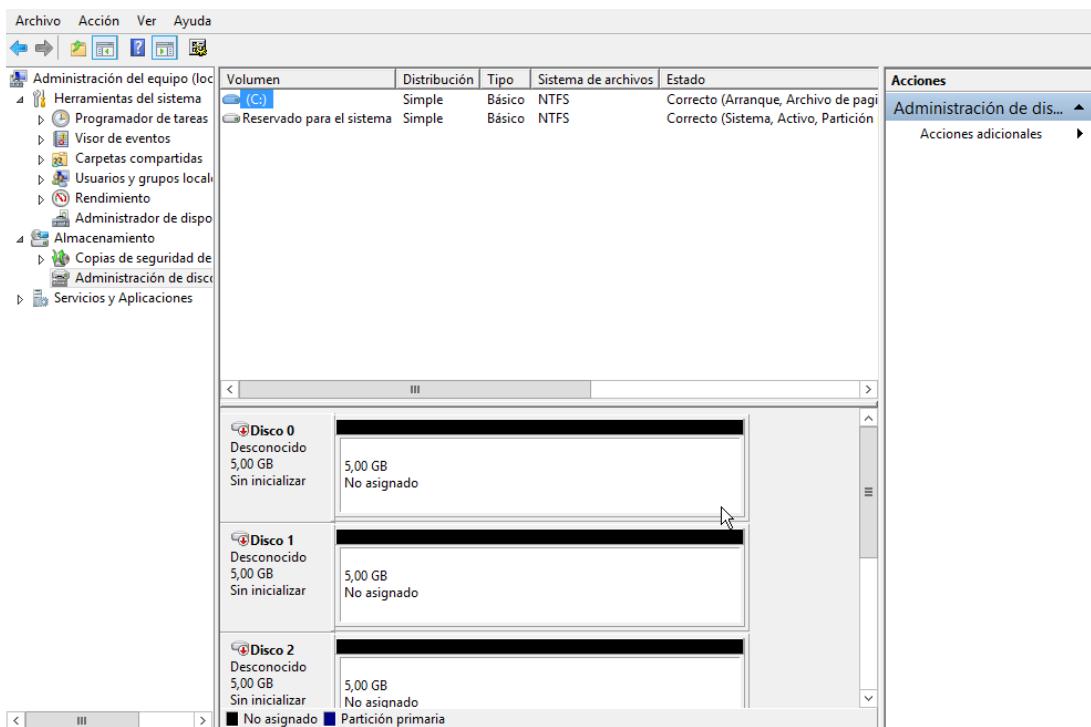
- Continuamos el asistente y seleccionamos custom para realizar la copia de seguridad solo de una carpeta determinada:



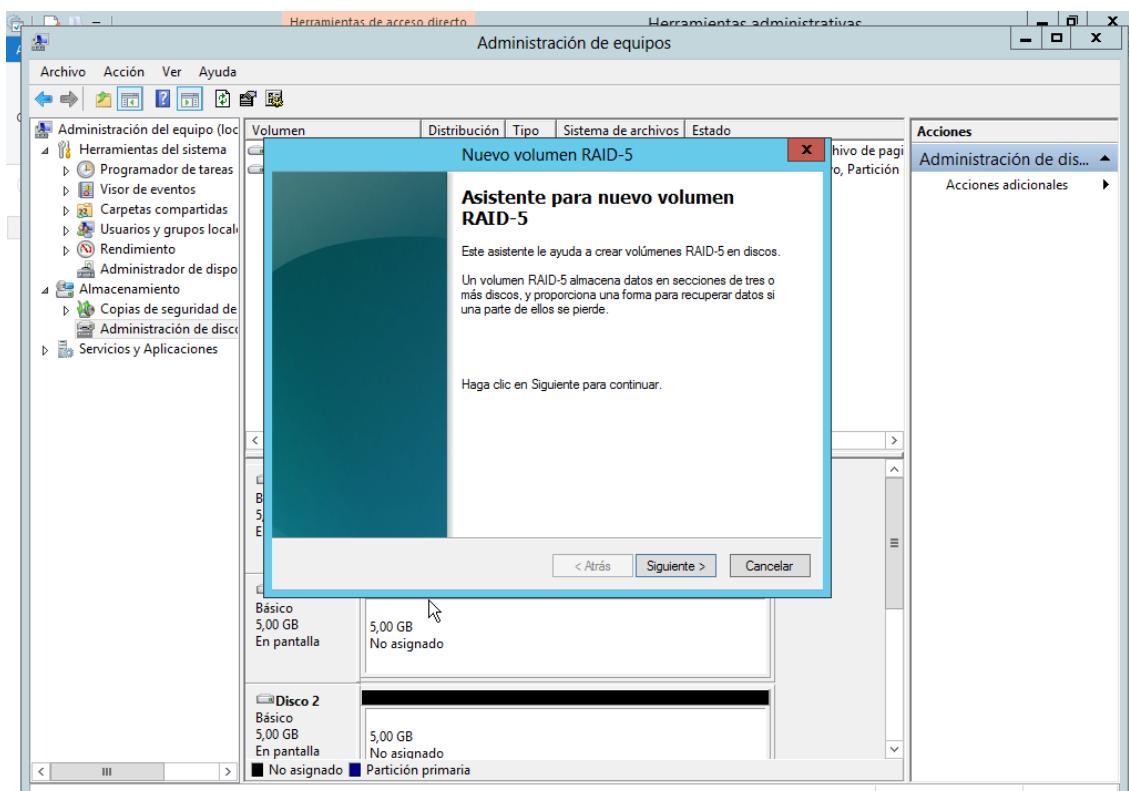
Donde la primera copia se realizará como completa y las sucesivas como incrementales.

## Raid

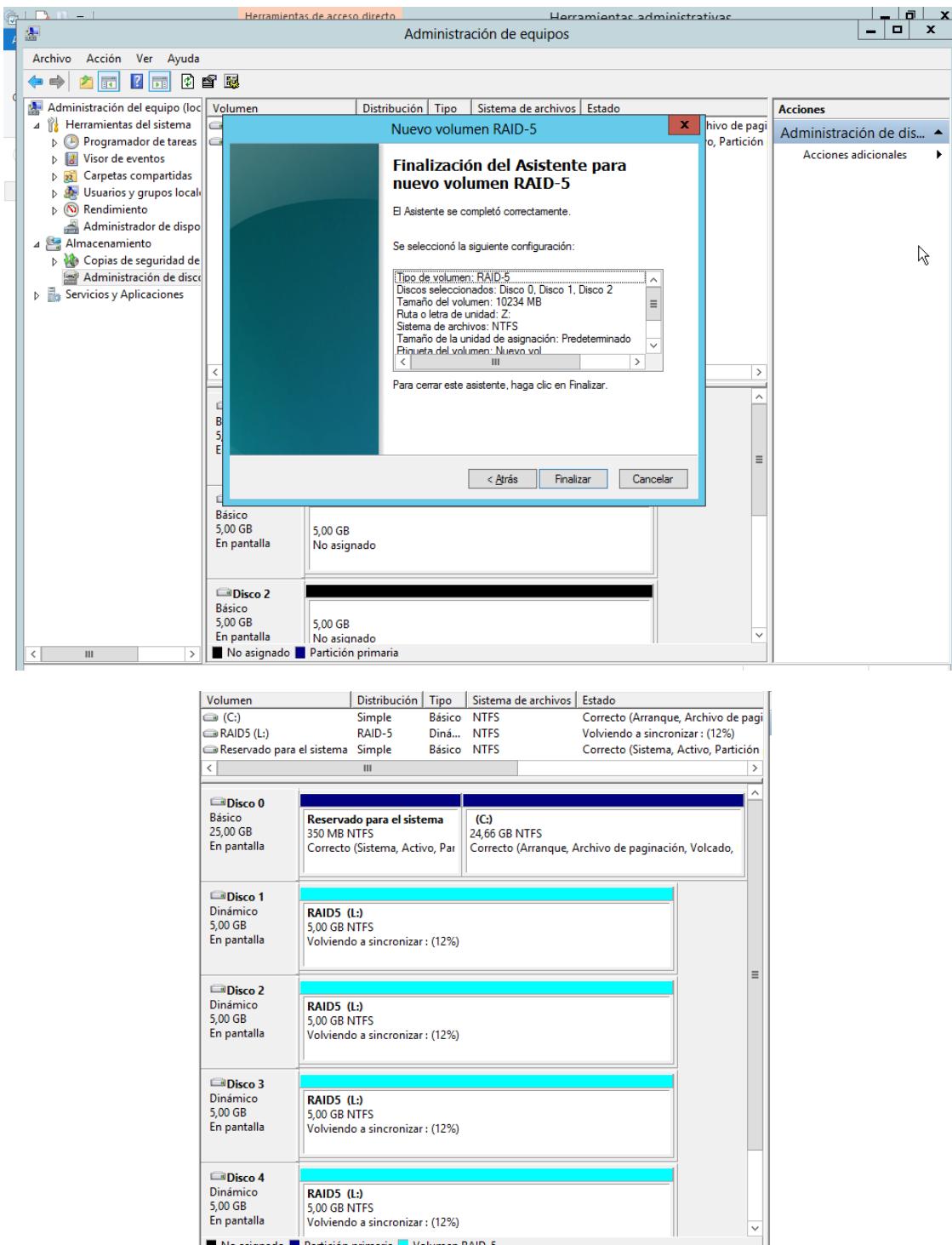
- Para la configuración de este servicio, dentro de la configuración de nuestra máquina virtual, agregamos 5 disco duros adicionales al que contiene nuestro SO, con el fin de que tres de ellos formen parte de nuestro RAID 5 y dos de un disco de RAID 1:



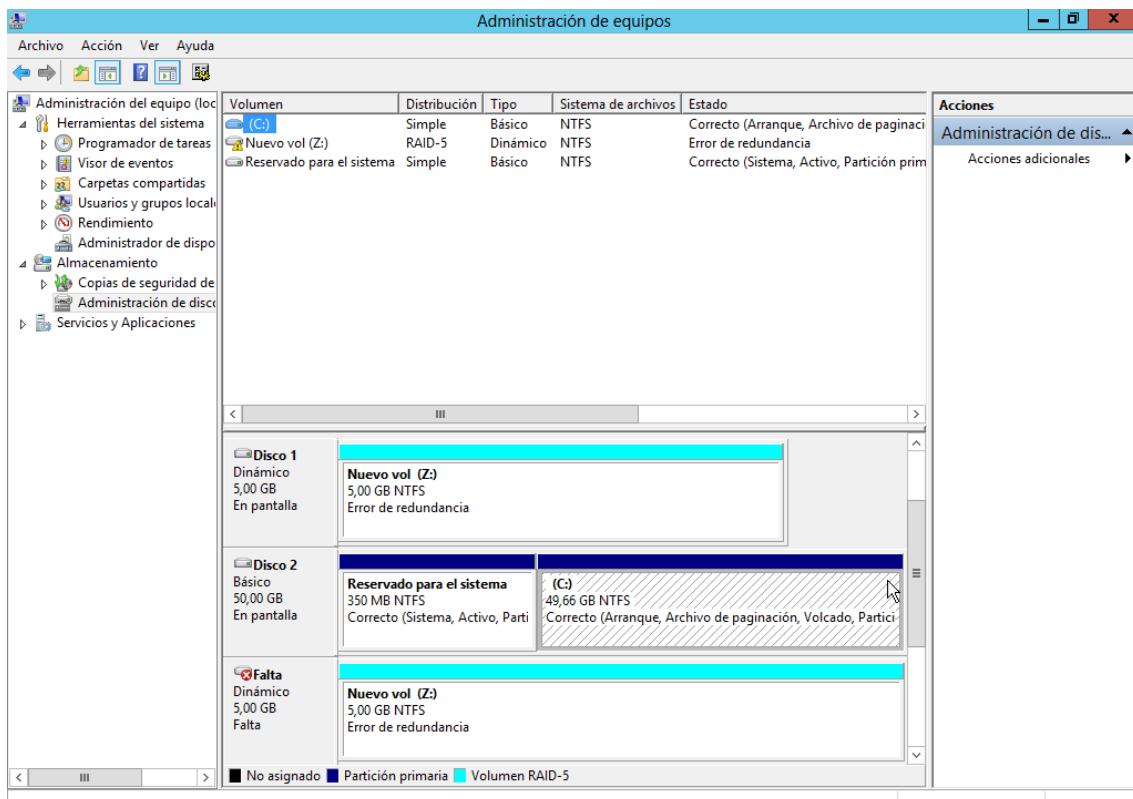
Al entrar en nuestro administrador de equipos, parte de almacenamiento, nos aparecerá el mensaje con los discos que desconocidos por el sistema y que se han detectado. Simplemente pinchando botón derecho e inicializando disco, seleccionamos los desconocidos. Y una vez realizado esto, seleccionando Asistente para RAID 1 ó 5 con el botón derecho, pasamos a la configuración de nuestro disco en RAID como se puede ver a continuación:



- Seleccionamos los discos que deseamos que formen nuestro RAID, y finalizamos para que comience la sincronización y creación de nuestro disco en RAID:



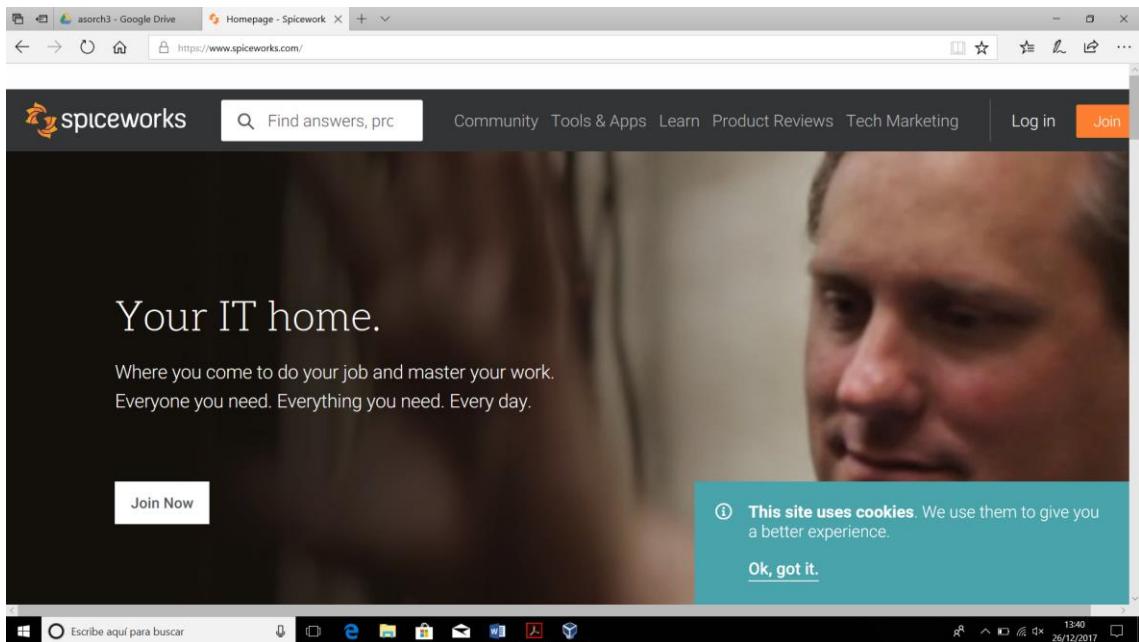
- Y para comprobar el correcto funcionamiento, saldríamos de la máquina virtual, desconectando un disco de los creados para nuestro RAID, obtendremos un error de redundancia, a pesar del cual, nuestro disco seguirá funcionando sin ningún tipo de problema, almacenando y permitiéndonos recuperar sin ningún tipo de problema la información deseada.



- Y bastaría con realizar la operación contraria, saliendo de la máquina y volviendo a conectar el disco desconectado anteriormente, para volver al administrador de equipos y mediante el botón derecho sobre el disco desconectado, seleccionando reactivar disco, pasaríamos a sincronizar de nuevo y seguir funcionando nuestros RAID sin ningún problema.

## Monitorización de servicios (Spiceworks)

- Para este servicio he elegido la configuración de **Spiceworks**, ya que Nagios no he podido instalarlo en Windows, como en casi todos los anteriores servicios, procedo a instalar **Spiceworks** entrando en su página principal y descargando:



- Configurando y seleccionando los servicios a monitorizar, podremos estar informados de nuestro correcto funcionamiento del servidor, como se puede observar a continuación:

