Instalar OpenCms desde 0

Tabla de contenido

[Version - 3 -](#_Toc468187163)

[Version 1.0 (27/10/2016) - 3 -](#_Toc468187164)

[Requerimientos - 3 -](#_Toc468187165)

[Red - 3 -](#_Toc468187166)

[Java - 4 -](#_Toc468187167)

[Tomcat 8 - 5 -](#_Toc468187168)

[Permisos y pertenencia a usuario y grupo - 5 -](#_Toc468187169)

[Tomcat como servicio - 5 -](#_Toc468187170)

[Configuración específica - 6 -](#_Toc468187171)

[Apache 2 - 9 -](#_Toc468187172)

[Módulos - 9 -](#_Toc468187173)

[Mod jk - 9 -](#_Toc468187174)

[Virtual Host - 11 -](#_Toc468187175)

[Sincronizar con OpenCms - 13 -](#_Toc468187176)

[1. Modificamos la plantilla de generación de virtual host para opencms - 13 -](#_Toc468187177)

[2. Cofiguramos permisos y pertenencia - 14 -](#_Toc468187178)

[3. Modificamos Apache para aceptar los virtual host creados por OpenCms - 15 -](#_Toc468187179)

[OpenCms - 15 -](#_Toc468187180)

[Instalación - 15 -](#_Toc468187181)

[Configuración - 15 -](#_Toc468187182)

[SMB - 16 -](#_Toc468187183)

[Linux - 16 -](#_Toc468187184)

[Windows - 17 -](#_Toc468187185)

[Configuración de nuevos repositorios - 17 -](#_Toc468187186)

[Configuración para eclipse - 18 -](#_Toc468187187)

# Versión

## Versión 1.0 (27/10/2016)

Documentamos el proceso para instalar OpenCms desde 0

# Requerimientos

Ref: <http://documentation.opencms.org/opencms-documentation/introduction/getting-opencms/#download>

* Ubuntu
  + Description: Ubuntu 16.04.1 LTS
  + Release: 16.04
  + Codename: xenial
  + Memoria al menos 2GB RAM
* Java 7: No superior al release 45 para evitar problema con Captcha del Webform.
* Tomcat 7 u 8. Configurado con memoria suficiente, mínimo 2 GB
* Apache 2.4

# Red

Editamos el fichero **/etc/network/interfaces**:

# This file describes the network interfaces available on your system and how to activate them. For more information, see interfaces(5).

#source /etc/network/interfaces.d/\*

# The loopback network interface

auto lo

iface lo inet loopback

# The primary network interface

auto enp0s3

iface enp0s3 inet static

address 192.168.1.80

gateway 192.168.1.1

netmask 255.255.255.0

network 192.168.1.0

broadcast 192.168.1.255

dns-nameservers 192.168.1.10 62.81.16.213

# Java

Ref: <https://www.digitalocean.com/community/tutorials/instalar-java-en-ubuntu-con-apt-get-es>

OpenCms recomienda instalar Java SDK, version 6 o mayor. Nosotros instalamos Java 7: No superior al release 45 para evitar problema con Captcha del Webform.

1. Agregar repositorio Java de Oracle

sudo apt-get install python-software-properties

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

1. Instalar java 7

sudo apt-get install oracle-java7-installer

# Tomcat 8

Ref: <https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>

Tomcat 7 u 8. Configurado con memoria suficiente, mínimo 2 GB.

1. Crear usuario tomcat

sudo groupadd tomcat

1. Configuramos al usuario miembro del grupo tomcat con directorio home /opt/tomcat y evitando que nadie pueda acceder a la máquina usandolo

sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

1. Descargamos tomcat

cd /tmp

curl -O http://apache.mirrors.ionfish.org/tomcat/tomcat-8/v8.5.5/bin/apache-tomcat-8.5.5.tar.gz

1. Descomprimimos en su carpeta

sudo mkdir /opt/tomcat

sudo tar xzvf apache-tomcat-8\*tar.gz -C /opt/tomcat --strip-components=1

## Permisos y pertenencia a usuario y grupo

1. Pertenencia al usuario y grupo tomcat

cd /opt/tomcat

sudo chgrp -R tomcat /opt/tomcat

sudo chown -R tomcat webapps/ work/ temp/ logs/

1. Configuramos permisos para carpetas y ficheros

sudo find /opt/tomcat/webapps/opencms10 -type d -exec chmod 755 {} +

sudo find /opt/tomcat/webapps/opencms10 -type f -exec chmod 644 {} +

## Tomcat como servicio

Ref: <https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units>

1. Obtenemos la versión de java instalada

sudo update-java-alternatives –l

1. Creamos el fichero tomcat.service

sudo nano /etc/systemd/system/tomcat.service

1. Con el siguiente contenido sustituyendo la veriable **JAVA\_HOME**

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre

Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

#Environment config in setenv.sh file

#Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

#Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

User=tomcat

Group=tomcat

#UMask=0755

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

1. Guardamos, iniciamos tomcat y comprobamos posibles fallos

sudo systemctl daemon-reload

sudo systemctl start tomcat

sudo systemctl status tomcat

## Configuración específica

1. Crear setenv

sudo nano /opt/tomcat/bin/setenv.sh

1. Configurar con el siguiente contenido:

JAVA\_OPTS="-Djava.awt.headless=true -Dfile.encoding=UTF-8 -server -Xms2048m -Xmx4096m -XX:NewSize=256m -XX:MaxNewSize=256m -XX:PermSize=256m -XX:MaxPermSize=256m -XX:+DisableExplicitGC"

# JAVA\_OPTS

# You could do all this in one export command

# But I am going to be long winded and explain

# Why and add links

#

# Oracle Notes

# - https://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html

# - A good visual

# <https://redstack.wordpress.com/2011/01/06/visualising-garbage-collection-in-the-jvm/>

#############################################

# -server

# Select the java HotSpot Server JVM

# The 64-bit version of the JDK support only the Server VM,

# so in that case the option is implicit

# ... so it s redundant to today s world but it make me feel good.

**export JAVA\_OPTS="-server"**

# -Xms/Xmx

# Xms Sets the initial size of the Heap

# Xmx sets the Maximum size of the Heap.

# http://stackoverflow.com/questions/16087153/what-happens-when-we-set-xmx-and-xms-equal-size

# http://crunchify.com/jvm-tuning-heapsize-stacksize-garbage-collection-fundamental/

**export JAVA\_OPTS="$JAVA\_OPTS -Xms2048m -Xmx4096m"**

# -NewSize/MaxNewSize

# Set the size of the young generation

# Most newly created objects are made here

# Objects taht did not become unreachbale and survice the young

# Generation heap are copied to the Old Generation

# See http://www.cubrid.org/blog/dev-platform/understanding-java-garbage-collection

# https://redstack.wordpress.com/2011/01/06/visualising-garbage-collection-in-the-jvm/

**export JAVA\_OPTS="$JAVA\_OPTS -XX:NewSize=256m -XX:MaxNewSize=256m"**

# -PermSize/MaxPermSize

# Store classes and interned character strings

# http://stackoverflow.com/questions/12114174/what-does-xxmaxpermsize-do

# Warning!

# Decprecated in Java 8!! replace -XX:MetaspaceSize !!!

**export JAVA\_OPTS="$JAVA\_OPTS -XX:PermSize=256m -XX:MaxPermSize=256m"**

# -UseConcMarkSweepGC

# Also called the low latency GC since pausing time is very short

# When this is enabled it also enabled

# -XX:+UseParNewGC Potentially speed up your generation GC

# by a factor equal to the number of CPUS

# (see http://stackoverflow.com/questions/2101518/difference-between-xxuseparallelgc-and-xxuseparnewgc)

# <http://www.cubrid.org/blog/dev-platform/understanding-java-garbage-collection/>

**export JAVA\_OPTS="$JAVA\_OPTS -XX:+UseConcMarkSweepGC"**

# -XX:+CMSIncrementalMode

#

# I am not going to set this one but it's worthe mentioning

# It has been deprecated in Java 8. It is useful when you have 1 or 2

# CPU machine. It helps reduce latency by doing smaller garbage collections

# see thies sites for details

# http://www.fixdeveloper.com/2014/03/jvm-tuning-cmsincrementalmode-overrides.html

# http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html#icms

# -CMSClassUnloadingEnabled

# In an old school java program classes are forever. But with

# Modern languages like Groovy... Classes are created at runtime, every

# scirpt may create a few new classes. With this set the PermGen space will

# Be garbage collecte3d. Without this you have a memory Leak.

#

# Must also have UseConcMarkSweepGC set for this to work.

#

# http://stackoverflow.com/questions/3334911/what-does-jvm-flag-cmsclassunloadingenabled-actually-do

**export JAVA\_OPTS="$JAVA\_OPTS -XX:+CMSClassUnloadingEnabled"**

# -DisableExlicitGC

# Explicit calls to System.gc() are completely ignored

#

# http://stackoverflow.com/questions/12847151/setting-xxdisableexplicitgc-in-production-what-could-go-wrong

**export JAVA\_OPTS="$JAVA\_OPTS -XX:+DisableExplicitGC"**

# -java.awt.headless

# Basically tell the JVM not to load awt libraries

# Your server is not a desktop app, there is more to this rule than that.

# If you want to go into it check out.

# https://blog.idrsolutions.com/2013/08/what-is-headless-mode-in-java/

# http://www.oracle.com/technetwork/articles/javase/headless-136834.html

**export JAVA\_OPTS="$JAVA\_OPTS -Djava.awt.headless=true"**

# Override Tomcat's default UMASK of 0027

**export UMASK="0022"**

# File encoding

**export JAVA\_OPTS="$JAVA\_OPTS -Dfile.encoding=UTF-8"**

1. Configuramos el session coockie path
2. Editamos el contexto

sudo nano /opt/tomcat/conf/context.xml

1. Agregamos la propiedad sessionCookiePath

<Context sessionCookiePath="/">

1. Configuramos server
2. Editamos el servidor

sudo nano /opt/tomcat/conf/server.xml

1. Agregando URIEncoding

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000" URIEncoding="UTF-8"

redirectPort="8443" />

1. Aseguramos que el Puerto 8009 está configurado

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" URIEncoding="UTF-8"/>

# Apache 2

1. Instalamos Apache 2

sudo apt-get update

sudo apt-get install apache2

1. Comprobamos el estado

sudo apache2ctl configtest

1. Configuramos apache con nuestro servidor
2. Comprobamos nuestra IP

curl http://icanhazip.com

1. Editamos el fichero de configuración de apache

sudo nano /etc/apache2/apache2.conf

1. Añadiendo la variable ServerName con el dominio o la IP

…

ServerName mv80.sagadev

## Módulos

Para que funcione correctamente es necesario tener instalados los siguientes módulos: **mod\_jk, mod\_proxy, mod\_proxy\_http, mod\_rewrite y mod\_deflate.**

Si utilizamos mod\_jk podemos eliminar **proxy\_ajp.**

## Mod jk

Ref: <https://tomcat.apache.org/tomcat-3.3-doc/mod_jk-howto.html>

Ref: <http://www.diegoacuna.me/installing-mod_jk-on-apache-httpd-in-centos-6-x7-x/>

1. Para instalar mod\_jk con apt-get

sudo apt-get install libapache2-mod-jk

1. Configuración del módulo disponible en /etc/apache2/mods-enabled/jk.conf
2. Asegurar el módulo **rewrite.load** existe en mods-enabled
3. Definimos **ocms** worker en /etc/libapache2-mod-jk/workers.properties

#

# workers.tomcat\_home should point to the location where you

# installed tomcat. This is where you have your conf, webapps and lib

# directories.

#

workers.tomcat\_home=/usr/share/tomcat8

#

# workers.java\_home should point to your Java installation. Normally

# you should have a bin and lib directories beneath it.

#

workers.java\_home=/usr/lib/jvm/default-java

#

# You should configure your environment slash... ps=\ on NT and / on UNIX

# and maybe something different elsewhere.

#

ps=/

#

#------ worker list ------------------------------------------

#---------------------------------------------------------------------

#

#

# The workers that your plugins should create and work with

#

#worker.list=ajp13\_worker

**worker.list=ocms**

#

#------ ocms WORKER DEFINITION ------------------------------

#---------------------------------------------------------------------

#

**worker.ocms.port=8009**

**worker.ocms.host=localhost**

**worker.ocms.type=ajp13**

#

#------ DEFAULT LOAD BALANCER WORKER DEFINITION ----------------------

#---------------------------------------------------------------------

#

#

# The loadbalancer (type lb) workers perform wighted round-robin

# load balancing with sticky sessions.

# Note:

# ----> If a worker dies, the load balancer will check its state

# once in a while. Until then all work is redirected to peer

# workers.

worker.loadbalancer.type=lb

**worker.loadbalancer.balance\_workers=ocms**

## Virtual Host

1. Creamos VH para el nuevo opencms

<VirtualHost \*:80>

DocumentRoot "/opt/tomcat/webapps/opencms10"

ServerName mv80.sagadev

ServerAdmin webmaster@mv80.sagadev

# Allow accessing the document root directory

<Directory /opt/tomcat/webapps/opencms10/resources>

Options FollowSymlinks

AllowOverride All

Order allow,deny

Allow from all

Require all granted

</Directory>

<Directory /opt/tomcat/webapps/opencms10/export>

Options FollowSymlinks

AllowOverride All

Order allow,deny

Allow from all

Require all granted

</Directory>

# Logging

LogFormat "%h %l %u %t \"%r\" %>s %b" common

CustomLog /var/log/apache2/access\_mv80.sagadev.log common

ErrorLog /var/log/apache2/error\_mv80.sagadev.log

LogLevel alert rewrite:trace1

# Do not forward the request if the requested URI is located in the resources folder or is static content

SetEnvIfNoCase Request\_URI ^/opencms10/resources/.\*$ no-jk

SetEnvIfNoCase Request\_URI ^/export/.\*$ no-jk

# Turn rewriting on and define the rules

RewriteEngine On

# Deny access to php files in order to prevent reading resources

RewriteCond %{REQUEST\_FILENAME} (.+)\.php(.\*)

RewriteRule (.\*) / [F]

# If the requested URI is NOT located in the resources folder and is not static:

# Prepend an /opencms10/opencms to everything that does not already starts with it

# and force the result to be handled by the next URI-handler ([PT]) (JkMount in this case)

RewriteCond %{REQUEST\_URI} !^/opencms10/workplace.\*$

RewriteCond %{REQUEST\_URI} !^/opencms10/VAADIN/.\*$

RewriteCond %{REQUEST\_URI} !^/opencms10/resources/.\*$

RewriteCond %{REQUEST\_URI} !^/export/.\*

RewriteRule !^/opencms10/opencms/(.\*)$ /opencms10/opencms%{REQUEST\_URI} [PT]

# These are the settings for static export. If the requested resource is not already

# statically exported create a new request to the opencms404 handler. This has to be

# a new request, because the current would not get through mod\_jk because of the "no-jk".

RewriteCond %{REQUEST\_URI} ^/export/.\*$

RewriteCond "%{DOCUMENT\_ROOT}%{REQUEST\_FILENAME}" !-f

RewriteCond "%{DOCUMENT\_ROOT}%{REQUEST\_FILENAME}/index\_export.html" !-f

RewriteRule .\* http://localhost:8080/opencms10/opencms/handle404?exporturi=%{REQUEST\_URI}&%{QUERY\_STRING} [P]

# If the request starts with /opencms10/resources, substitute the /opencms10 prefix

RewriteCond %{REQUEST\_URI} ^/opencms10/resources/.\*$

RewriteRule ^/opencms10/(.\*)$ /$1

JkMount /\* ocms

</VirtualHost>

## Sincronizar con OpenCms

Vamos a sincronizar Apache con OpenCms para que al generar un site se cargue un nuevo VH automáticamente.

### Modificamos la plantilla de generación de virtual host para opencms

<VirtualHost \*:80>

DocumentRoot "$DOCUMENT\_ROOT$"

ServerName $SERVER\_NAME\_WITH\_PORT$

$ALIAS\_DIRECTIVE$ $SERVER\_ALIASES$

ServerAdmin webmaster@$SERVER\_NAME\_WITH\_PORT$

# Allow accessing the document root directory

<Directory $DOCUMENT\_ROOT$/resources>

Options FollowSymlinks

AllowOverride All

Order allow,deny

Allow from all

Require all granted

</Directory>

<Directory $DOCUMENT\_ROOT$/export>

Options FollowSymlinks

AllowOverride All

Order allow,deny

Allow from all

Require all granted

</Directory>

# Logging

LogFormat "%h %l %u %t \"%r\" %>s %b" common

CustomLog $LOGGING\_DIRECTORY$access\_$CONFIG\_FILENAME$.log common

ErrorLog $LOGGING\_DIRECTORY$error\_$CONFIG\_FILENAME$.log

LogLevel alert rewrite:trace1

# Do not forward the request if the requested URI is located in the resources folder or is static content

SetEnvIfNoCase Request\_URI ^$CONTEXT\_PATH$/resources/.\*\$ no-jk

SetEnvIfNoCase Request\_URI ^/export/.\*\$ no-jk

# Turn rewriting on and define the rules

RewriteEngine On

# Deny access to php files in order to prevent reading resources

RewriteCond %{REQUEST\_FILENAME} (.+)\.php(.\*)

RewriteRule (.\*) / [F]

# If the requested URI is NOT located in the resources folder and is not static:

# Prepend an $CONTEXT\_PATH$$SERVLET\_PATH$ to everything that does not already starts with it

# and force the result to be handled by the next URI-handler ([PT]) (JkMount in this case)

RewriteCond %{REQUEST\_URI} !^$CONTEXT\_PATH$/workplace.\*\$

RewriteCond %{REQUEST\_URI} !^$CONTEXT\_PATH$/VAADIN/.\*\$

RewriteCond %{REQUEST\_URI} !^$CONTEXT\_PATH$/resources/.\*\$

RewriteCond %{REQUEST\_URI} !^/export/.\*

RewriteRule !^$CONTEXT\_PATH$$SERVLET\_PATH$/(.\*)\$ $CONTEXT\_PATH$$SERVLET\_PATH$%{REQUEST\_URI} [PT]

# These are the settings for static export. If the requested resource is not already

# statically exported create a new request to the opencms404 handler. This has to be

# a new request, because the current would not get through mod\_jk because of the "no-jk".

RewriteCond %{REQUEST\_URI} ^/export/.\*\$

RewriteCond "%{DOCUMENT\_ROOT}%{REQUEST\_FILENAME}" !-f

RewriteCond "%{DOCUMENT\_ROOT}%{REQUEST\_FILENAME}/index\_export.html" !-f

RewriteRule .\* http://localhost:8080$CONTEXT\_PATH$$SERVLET\_PATH$/handle404?exporturi=%{REQUEST\_URI}&%{QUERY\_STRING} [P]

# If the request starts with $CONTEXT\_PATH$/resources, substitute the $CONTEXT\_PATH$ prefix

RewriteCond %{REQUEST\_URI} ^$CONTEXT\_PATH$/resources/.\*\$

RewriteRule ^$CONTEXT\_PATH$/(.\*)\$ /\$1

JkMount /\* ocms

</VirtualHost>

* !^$CONTEXT\_PATH$/workplace.\*\$ y !^$CONTEXT\_PATH$/VAADIN/.\*\$ solo son aplicables para **opencms10**

### Cofiguramos permisos y pertenencia

* 1. Aseguramos los permisos y la pertenencia del fichero **vhost.template**:

sudo chmod 644 vhost.template

sudo chown tomcat:tomcat vhost.template

* 1. Aseguramos los permisos de ejecución para el fichero script.sh

sudo chmod 777 script.sh

### Modificamos Apache para aceptar los virtual host creados por OpenCms

* 1. En Apache añadimos un link a la carpeta de generación de virtual host de OpenCms:

cd /etc/apache2/sites-enabled/

sudo ln -s /opt/tomcat/webapps/opencms10/WEB-INF/server-scripts/configs/

* 1. Editamos el fichero apache2.conf

sudo nano /etc/apache2/apache2.conf

* 1. Añadiendo la línea:

# Include the opencms virtual host:

IncludeOptional sites-enabled/configs/opencms\*

# OpenCms

## Instalación

1. Según la versión que queramos la descargamos de <http://www.opencms.org/en/download/opencms.html>
2. Debemos descomprimimos el zip dentro de webapps

unzip /tmp/opencms.zip –d /opt/tomcat/webapps/

1. Reiniciamos tomcat e instalamos OpenCms accediendo a <http://localhost:8080/opencms/opencms/setup>

## Configuración

1. Hacer login y cambiar la contraseña del usuario Admin.
2. Borrar la carpeta $TOMCAT$\webapps\opencms\**setup** y el war $TOMCAT$\webapps\**opencms.war**
3. Editar el fichero $TOMCAT$\webapps\opencms\WEB-INF\config\**opencms-system.xml** y agregar el **locale es** a la lista de locales disponibles en primera posición, tanto en <**localesconfigured**> como en <**localesdefault**>
4. Editar el fichero $TOMCAT$\webapps\opencms\WEB-INF\config\**opencms-system.xml** y configura el servidor de correos en el nodo <**mail**>
5. Editar el fichero $TOMCAT$\webapps\opencms\WEB-INF\config\**opencms-system.xml configurando el site default mv80**
6. Editar el fichero $TOMCAT$\webapps\opencms\WEB-INF\config\**opencms-workplace.xml** y cambiar el locale por defecto a **es**    
   <**defaultlocale>es</defaultlocale**>
7. Editar el fichero $TOMCAT$\webapps\opencms\WEB-INF\config\**opencms-importexport.xml** y modificar la exportación a disco y el prefijo de la URL:

<rendersettings>

<rfs-prefix>export</rfs-prefix>

<vfs-prefix></vfs-prefix>

</rendersettings>

1. Editar el fichero $TOMCAT$\webapps\opencms\WEB-INF\solr\conf\**schema.xml** y agregar el filtro **solr.ASCIIFoldingFilterFactory** a la configuración de tipo de campo **text\_general** para ignarar las tildes tanto en las búsquedas como en la indexación:   
   **<filter class="solr.ASCIIFoldingFilterFactory"/>**
2. En el mismo fichero de antes **schema.xml**, añade la configuración del tag como dynamicField:   
   **<dynamicField name="\*\_tag"        type="string"       indexed="true"  stored="true" multiValued="true"/>**

## SMB

### Linux

Ref: <http://www.opencmshispano.com/blog/detalle/Conectar-Eclipse-con-OpenCms-mediante-servidor-SMB>

Ref: <http://documentation.opencms.org/opencms-documentation/mounting-the-vfs/network-share-access/>

Ref: <https://help.ubuntu.com/community/IptablesHowTo#Saving_iptables>

Para poder sincronizar el contenido del VFS por samba:

1. Renombrar el fichero de configuración jLan según el sistema operativo. Nosotros seguiremos el ejemplo de Linux.

sudo mv jlanConfig.xml.linux jlanConfig.xml

1. Redirigimos el puerto 445 a 1445 con iptables, guardamos la configuración y la almacenamos en un fichero para posteriormente ejecutarla al arranque.

sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 445 -j REDIRECT --to-port 1445

sudo iptables-save

1. Comprobamos la configuración de red

sudo netstat –plntu

Debe ser algo como es muestra a continuación:

Conexiones activas de Internet (solo servidores)

Proto Recib Enviad Dirección local Dirección remota Estado PID/Program name

tcp 0 0 0.0.0.0:22 0.0.0.0:\* ESCUCHAR 954/sshd

tcp6 0 0 127.0.0.1:8005 :::\* ESCUCHAR 6263/java

**tcp6 0 0 :::1445 :::\* ESCUCHAR 6263/java**

**tcp6 0 0 :::8009 :::\* ESCUCHAR 6263/java**

**tcp6 0 0 :::8080 :::\* ESCUCHAR 6263/java**

**tcp6 0 0 :::80 :::\* ESCUCHAR 1123/apache2**

tcp6 0 0 :::22 :::\* ESCUCHAR 954/sshd

1. Cargamos la configuración al arranque
   1. Almacenamos la configuración en iptables.rules

sudo sh -c "iptables-save > /etc/iptables.rules"

* 1. Editamos interfaces

sudo nano /etc/network/interfaces

* 1. Añadimos al final de la configuración la instrucción pre-up

# The primary network interface

auto enp0s3

iface enp0s3 inet static

address 192.168.1.81

gateway 192.168.1.1

netmask 255.255.255.0

network 192.168.1.0

broadcast 192.168.1.255

dns-nameservers 192.168.1.10 62.81.16.213

**pre-up iptables-restore < /etc/iptables.rules**

### Windows

1. Utilizamos el fichero jlanConfig.xml.windows y le llamamos: jlanConfig.xml.
2. Ahora tenemos que deshabilitar el servicio de Windows llamado: "Server" o "Servidor"
3. Damos de alta en el fichero opencms-importexport.xml un nuevo repositorio:

<repository name="OPENCMS\_MODULES" class="org.opencms.jlan.CmsJlanRepository">

<params>

<param name="project">Offline</param>

<param name="root">/system/modules</param>

<param name="addBOM">false</param>

<param name="wrapper">org.opencms.file.wrapper.CmsResourceWrapperReplaceDeleted</param>

</params>

</repository>

1. Reiniciamos el servidor para que se actualice la configuración

### Configuración de nuevos repositorios

Como alternativa podemos crear nuevos repositorios a parte del repositorio por defecto, para ello damos de alta en el fichero opencms-importexport.xml el nuevo repositorio:

<repository name="OPENCMS\_MODULES" class="org.opencms.jlan.CmsJlanRepository">

<params>

<param name="project">Offline</param>

<param name="root">/system/modules</param>

<param name="addBOM">false</param>

<param name="wrapper">org.opencms.file.wrapper.CmsResourceWrapperReplaceDeleted</param>

</params>

</repository>

### Configuración para eclipse

Una vez que ya tengamos activado el nuevo repositorio que hemos llamado OPENCMS\_MODULES, podemos conectarnos a la nueva unidad de red. Para probar que todo está funcionando correctamente, intentamos abrir la carpeta usando la dirección:

\\IP\_SERVIDOR\OPENMCS\_MODULE\

Si todo ha funcionado correctamente deberás ver todas las carpetas correspondiente a los módulos de OpenCms.

Bien, ahora que tenemos esto, vamos a ver los pasos para conectar nuestro módulo en eclipse:

1. Nos creamos un nuevo proyecto de eclipse (Opcional, si ya tenéis uno creado podéis reutilizarlo)
2. Seleccionamos el proyecto donde vamos a añadir la carpeta y le damos a: File -> New -> Folder
3. Pulsamos el botón de Advanced >>
4. Marcamos la opción: Link to alternative location (Linked Folder)
5. En la casilla inferior añadimos la ruta a nuestro módulo, que será algo como: \\IP\_SERVIDOR\OPENMCS\system\modules\module\_name\. Es recomendable crear una unidad de red propia para el OpenCms y utilizar dicha unidad de red a la hora de sincronizarlo con OpenCms.
6. Opcionalmente podemos cambiar el nombre a la carpeta a crear.
7. Pulsamos botón Finish

Una vez realizado estos pasos ya tenedremos sincronizada una carpeta de OpenCms en nuestro eclipse, y lista para poder editar cualquier recurso dentro de ella. Todos los cambios que hagamos se guardarán automáticamente sobre el recurso de OpenCms sin necesidad de hacer nada más.