



---

# EJERCICIOS TEMA 2

---

INSTALACIÓN, CONFIGURACIÓN Y DOCUMENTACIÓN DEL ENTORNO DE  
DESARROLLO Y DEL ENTORNO DE EXPLOTACIÓN



7 DE OCTUBRE DE 2022  
ALEJANDRO OTALVARO MARULANDA

## Contenido

Instalación y configuración inicial (Plugins).....	2
Instalación y configuración.....	7
Instalación del módulo PHP.....	9
Instalación de Apache Tomcat.....	11
Instalación de Tomcat en Ubuntu Server.....	13
Instalación MYSQL.....	16
Configurar archivo .htaccess.....	18
Instalar Servidor DNS.....	19
Instalación HTTPS.....	24
Autenticación de acceso.....	27
DIGEST.....	27

# WXED-WINDOWS 10

## Instalación y configuración inicial (Plugins).

Para poder comenzar con la instalación de NetBeans, es necesario instalar antes el JDK de JAVA, para ello utilizaremos la versión 19 (JDK-19) para Windows 10 x64.

Daremos simplemente a confirmar la ubicación de instalación y ya acabaría.

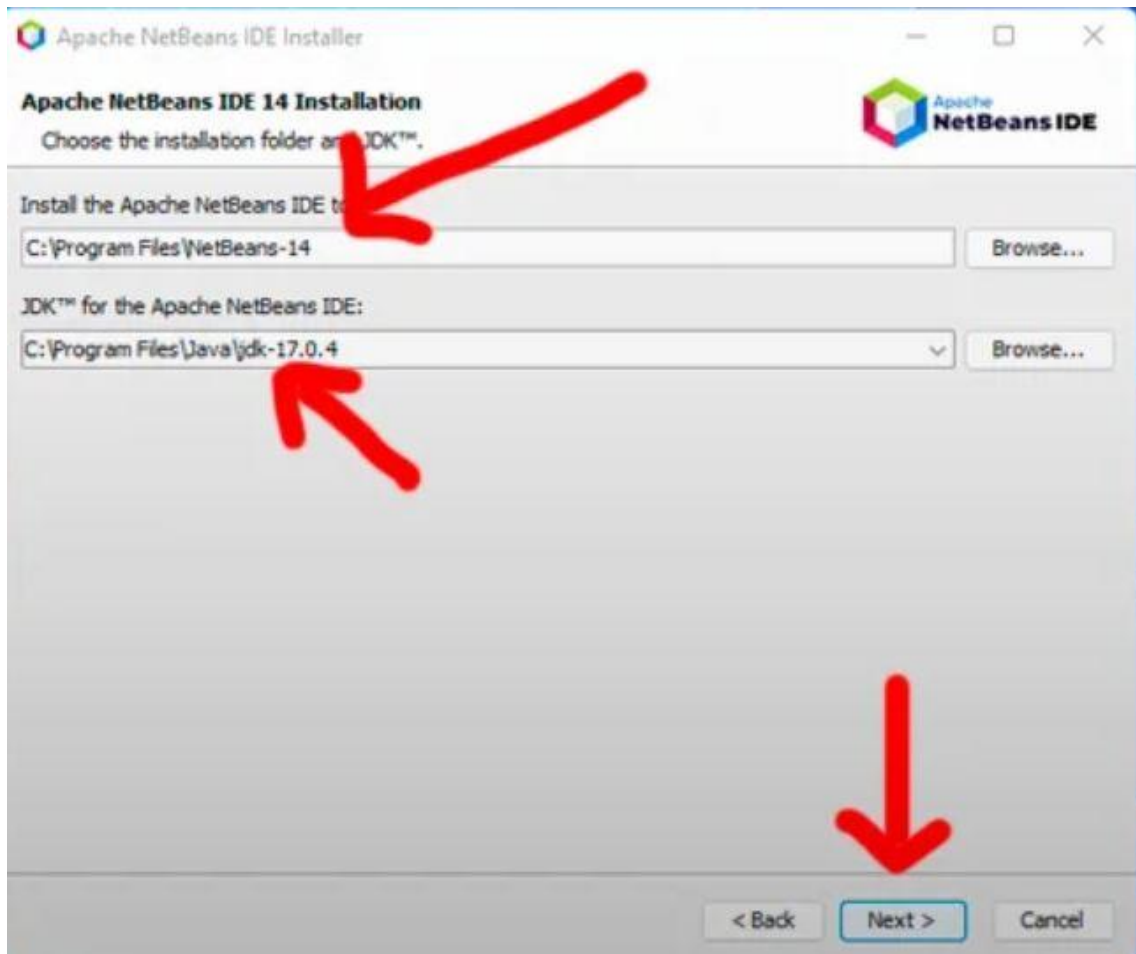
Instalaremos nuestro cliente FTP que en este caso será Filezilla. Para ello confirmamos ubicación y se instalará.



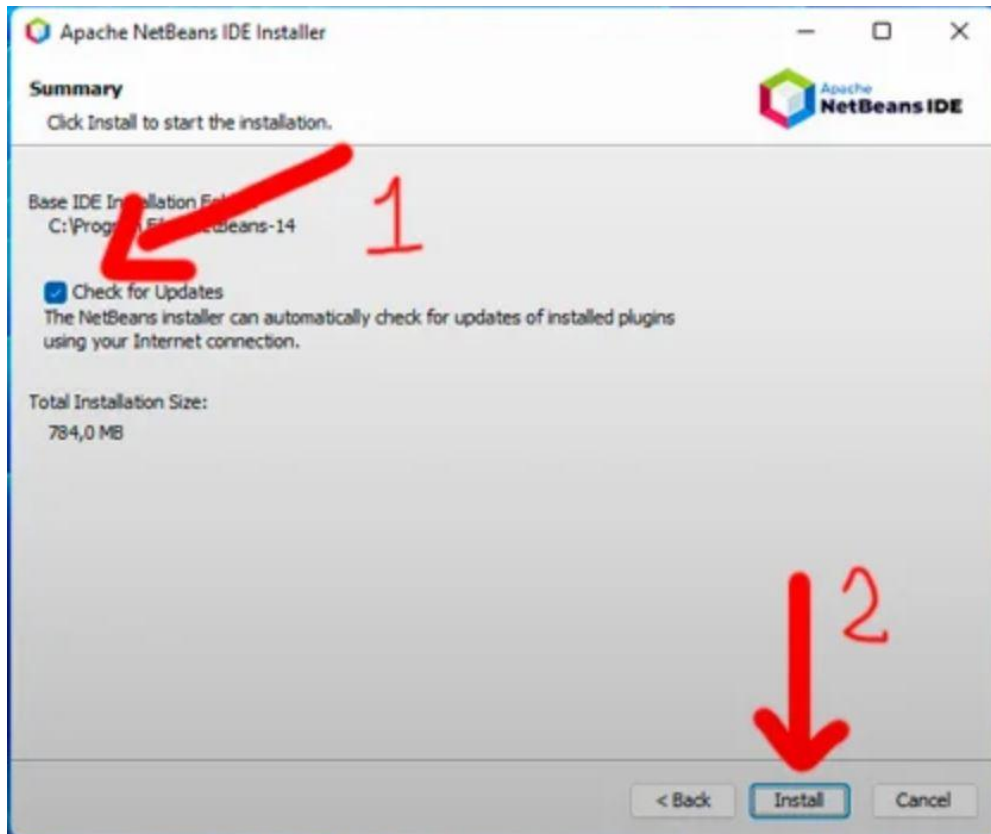
Hecho esto, procederemos a la instalación de NetBeans:

1. Aceptaremos los términos de licencia.

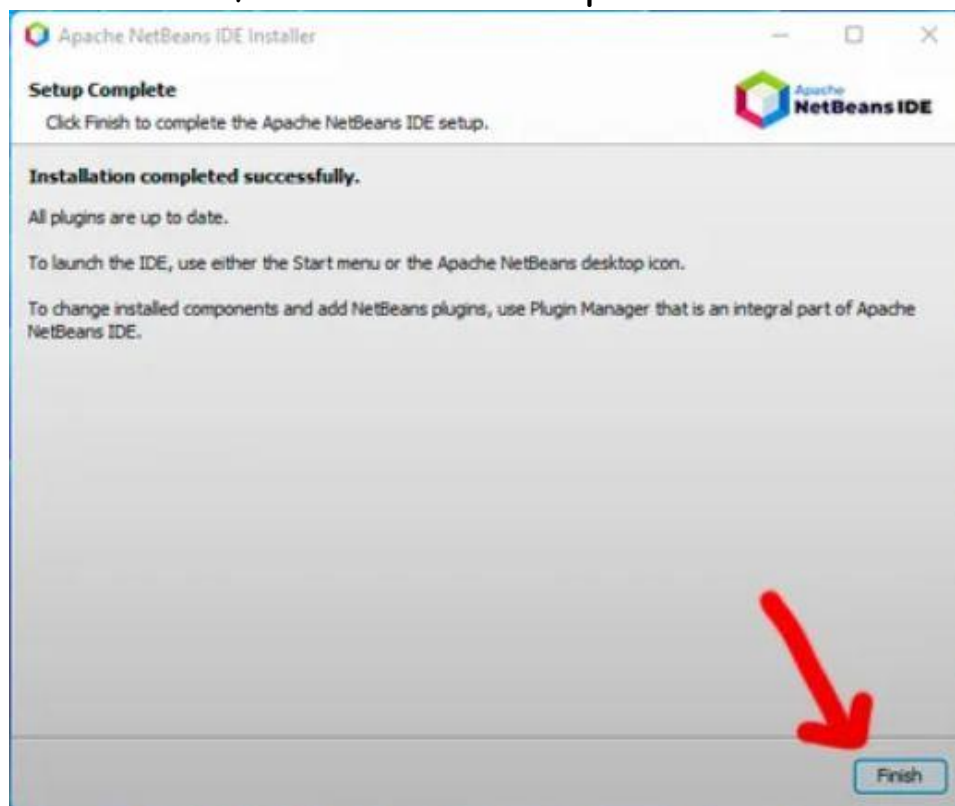
2. Escogeremos la ubicación de la instalación y la ubicación del JDK.



3. Después le daremos para que actualice a la ultima versión si esa no es la descargada.



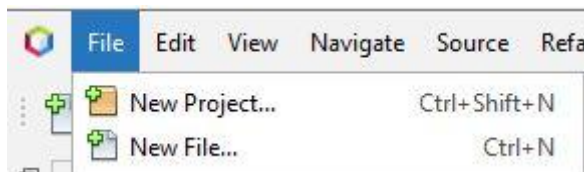
4. Hecho esto, finalizaremos el proceso de instalación



Instalaremos un navegador para comprobar nuestros avances en el Lenguaje de Programación *PHP*.



Para la creación de un nuevo proyecto en NetBeans, iremos a la pestaña de File y pulsaremos en New Project



Cuando creamos un nuevo proyecto en NetBeans, se crean una serie de carpetas y archivos por defecto.

nbproject	29/09/2022 9:56	Carpeta de archivos
public_html	03/10/2022 9:40	Carpeta de archivos
test	29/09/2022 9:56	Carpeta de archivos
.bowerrc	29/09/2022 9:56	Archivo BOWERRC
bower.json	29/09/2022 9:56	Archivo JSON
CV	29/09/2022 10:11	Archivo JPEG
Gruntfile	29/09/2022 9:56	Archivo JavaScript
gulpfile	29/09/2022 9:56	Archivo JavaScript
package.json	29/09/2022 9:56	Archivo JSON

En la cual, nuestro index.html lo tendremos en la carpeta que se llama, public\_html.

# USED-UBUNTU SERVER

## 22

### Instalación y configuración

Para la instalación del sistema Operativo hemos usado una imagen de Ubuntu Server con la versión: ubuntu-22.04.1-live-server-amd64.

Durante la configuración del sistema, escribiremos el nombre del equipo que queramos o el que se nos implante en el proyecto.

En dicha instalación, crearemos el usuario miadmin cuyo password será paso.

También crearemos un usuario llamado operador web, para ello usaremos el comando:

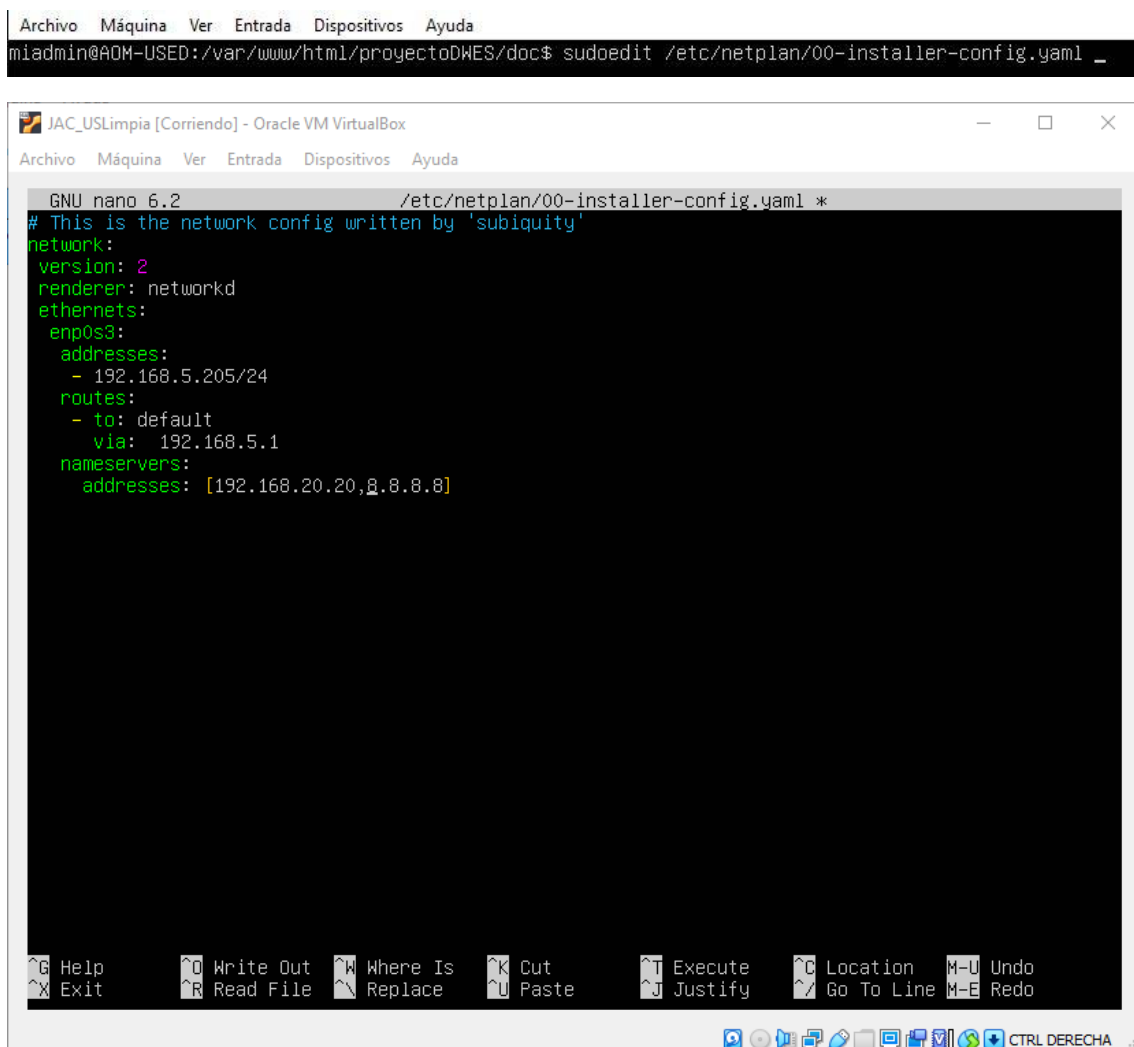


*"sudo adduser operador web".*

Este usuario será el encargado de desarrollar nuestras aplicaciones

Hecho esto, procederemos a configurar la red, en este caso para que esté en la red local de la clase.

Para ello configuraremos el archivo usando este comando:



```
miadmin@AOM-USED:/var/www/html/proyectoDWES/doc$ sudoedit /etc/netplan/00-installer-config.yaml _
```

```
GNU nano 6.2 /etc/netplan/00-installer-config.yaml *
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 192.168.5.205/24
      routes:
        - to: default
          via: 192.168.5.1
      nameservers:
        addresses: [192.168.20.20,8.8.8.8]
```

Cuando ya tenemos la red configurada, procederemos a la instalación del servicio SSH, para ello usaremos el comando:

***sudo ufw enable***

Hecho esto, usaremos el comando ***sudo ufw allow 22/tcp*** para abrir el puerto 22.

A continuación, instalaremos el servicio apache usando el comando:

***sudo apt install apache2***

Usaremos el mismo comando para abrir el puerto 22 para el puerto 80, para que se puedan conectar para ver nuestro sitio web.

## Instalación del módulo PHP

Para la instalación de PHP, ejecutaremos el comando:

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo apt install php
```

Confirmamos que deseamos descargar todo:

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo apt install php  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  libapache2-mod-php8.1 php-common php8.1 php8.1-cli php8.1-common  
  php8.1-opcache php8.1-readline  
Paquetes sugeridos:  
  php-pear  
Se instalarán los siguientes paquetes NUEVOS:  
  libapache2-mod-php8.1 php php-common php8.1 php8.1-cli php8.1-common  
  php8.1-opcache php8.1-readline  
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 17 no actualizados.  
Se necesita descargar 5.126 kB de archivos.  
Se utilizarán 21,3 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n]
```

Al instalar, comprobamos la versión de php con el comando:

```
miadmin@AOM-USED:~$ php -v  
PHP 8.1.2 (cli) (built: Aug 8 2022 07:28:23) (NTS)  
Copyright (c) The PHP Group  
Zend Engine v4.1.2, Copyright (c) Zend Technologies  
with Zend OPcache v8.1.2, Copyright (c), by Zend Technologies  
miadmin@AOM-USED:~$
```

Vamos a configurar el archivo de configuración:

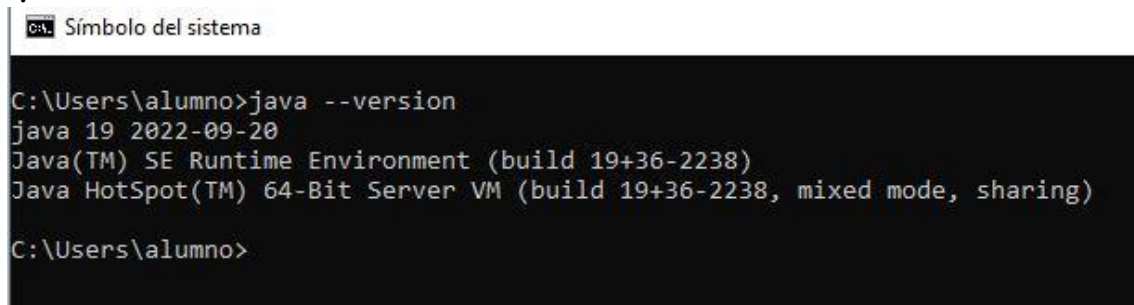
```
miadmin@AOM-USED:/$ sudoedit php.ini
```

Cuando cambiamos las directivas, reiniciamos el servicio Apache:

```
miadmin@AOM-USED:/$ sudo service apache2 restart
```

# Instalación de Apache Tomcat

Empezamos comprobando que tenemos java instalado y que versión tenemos:

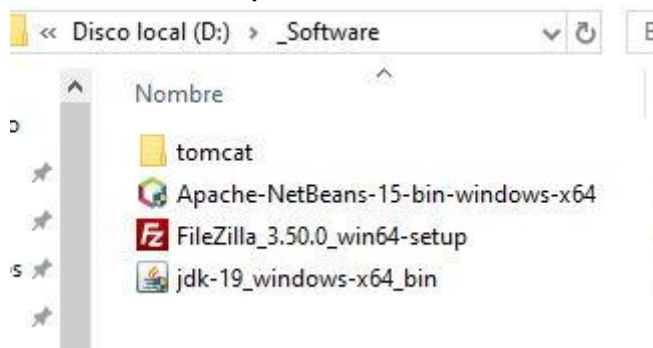


```
C:\Users\alumno>java --version
java 19 2022-09-20
Java(TM) SE Runtime Environment (build 19+36-2238)
Java HotSpot(TM) 64-Bit Server VM (build 19+36-2238, mixed mode, sharing)
C:\Users\alumno>
```

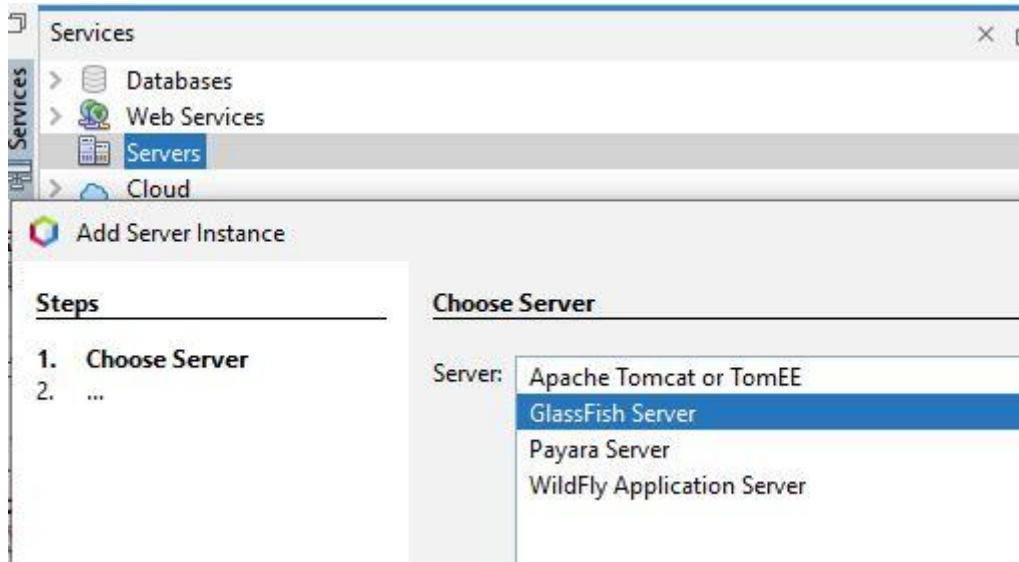
Después, en la página principal de tomcat, descargaremos la versión que queremos. En nuestro caso: 10.0.27

<https://tomcat.apache.org/download-10.cgi#10.0.27>

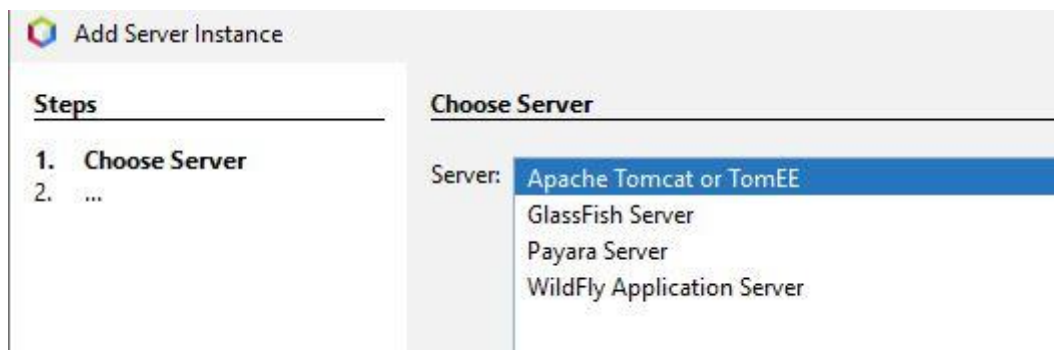
Descargaremos un zip y lo meteremos en la carpeta \_Software que tenemos en nuestro disco D:\



Crearemos un server en Netbeans:



Escogemos Apache Tomcat or TomEE



# Instalación de Tomcat en Ubuntu Server

Para ello, el primer paso es actualizar el sistema:

*sudo apt update*

*sudo apt upgrade*

Buscaremos la versión 18:

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo apt-cache search openjdk  
default-jdk - Kit de desarrollo de Java estándar o Java compatible  
default-jdk-doc - Conjunto de herramientas de desarrollo compatibles con  
JDK) (documentación)
```

Después, instalaremos JDK:

```
miadmin@AOM-USED:~$ sudo apt install openjdk-18-jdk  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho
```

Una vez instalado, comprobaremos que nos lo ha instalado bien:

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ java --version  
openjdk 18.0.2-ea 2022-07-19  
OpenJDK Runtime Environment (build 18.0.2-ea+9-Ubuntu-222.04)  
OpenJDK 64-Bit Server VM (build 18.0.2-ea+9-Ubuntu-222.04, mixed mode, sharing)  
miadmin@AOM-USED:~$
```

Ahora, instalaremos Tomcat:

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo apt install tomcat9 tomcat9-admin
```

Abrimos el puerto 8080 para TOMCAT

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo ufw allow 8080/tcp  
Rule added  
Rule added (v6)  
miadmin@AOM-USED:~$
```

Y lo comprobamos

```
miadmin@AOM-USED:~$ sudo ufw status  
Status: active  
  
To Action From  
--  
22/tcp ALLOW Anywhere  
80 ALLOW Anywhere  
9003 ALLOW Anywhere  
8080/tcp ALLOW Anywhere  
22/tcp (v6) ALLOW Anywhere (v6)  
80 (v6) ALLOW Anywhere (v6)  
9003 (v6) ALLOW Anywhere (v6)  
8080/tcp (v6) ALLOW Anywhere (v6)  
miadmin@AOM-USED:~$
```

Hecho esto, procedemos a configurar el usuario tomcat.

Para ello, editaremos el fichero:

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo nano /etc/tomcat9/tomcat-users.xml
```

Y añadiremos FUERA DE LOS COMENTARIOS, las siguientes instrucciones:

```
<role rolename="admin-gui"/>
<role rolename="manager-gui"/>
<user username="tomcat" password="paso" roles="admin-gui,manager-gui"/>

</tomcat-users>
```

Hecho esto, reiniciamos el servicio de Tomcat:

```
miadmin@AOM-USED: ~
miadmin@AOM-USED:~$ sudo systemctl restart tomcat9.service
```



# Instalación MYSQL

Procederemos a ejecutar el comando:

```
miadmin@AOM-USED:~$ sudo apt install mysql-server
```

Hecho esto, comprobaremos la versión del paquete instalado:

```
miadmin@AOM-USED:~$ dpkg -l mysql-server
Desead=desconocido(U)/Instalar/eliminar/Purgar/retener(H)
| Estado=No/Inst/ficheros-Conf/desempaquetado/medio-conf/medio-inst(H)/espera-disparo(W)/pendiente-disparo
|/ Err?=(ninguno)/requiere-Reinst (Estado,Err: mayúsc.=malo)
||/ Nombre Versión Arquitectura Descripción
+++-----+-----+-----+-----+
ii mysql-server 8.0.31-0ubuntu0.22.04.1 all MySQL database server (metapackage depending on the latest version)
miadmin@AOM-USED:~$
```

Comprobamos el estado de mysql:

```
miadmin@AOM-USED:~$ sudo service mysql status
• mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-11-03 09:29:37 UTC; 2min 43s ago
     Process: 2423 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 2431 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 2238)
       Memory: 363.4M
          CPU: 2.362s
      CGroup: /system.slice/mysql.service
              └─2431 /usr/sbin/mysqld

nov 03 09:29:36 AOM-USED systemd[1]: Starting MySQL Community Server...
nov 03 09:29:37 AOM-USED systemd[1]: Started MySQL Community Server.
miadmin@AOM-USED:~$
```

Miraremos los puertos que usa mysql: en este caso 3306

```
miadmin@AOM-USED:~$ ss -ltn
```

Abrimos el puerto 3306:

```
miadmin@AOM-USED:~$ sudo ufw allow 3306
Rule added
Rule added (v6)
miadmin@AOM-USED:~$
```

Crear usuarios y darle los privilegios:

```
create user 'adminsql'@'%'identified by 'paso';
```

```
grant all privileges en *.* to 'adminsql'@'%' with grant option;
```

```
miadmin@AOM-USED:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
80 ALLOW Anywhere
9003 ALLOW Anywhere
8080/tcp ALLOW Anywhere
9003/tcp ALLOW Anywhere
9000 DENY Anywhere
3306 ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
9003 (v6) ALLOW Anywhere (v6)
8080/tcp (v6) ALLOW Anywhere (v6)
9003/tcp (v6) ALLOW Anywhere (v6)
9000 (v6) DENY Anywhere (v6)
3306 (v6) ALLOW Anywhere (v6)

miadmin@AOM-USED:~$
```

Para instalar el conector de mysql en el servidor escribimos:

```
miadmin@AOM-USED:~$ sudo apt install php8.1-mysql
```

# Configurar archivo .htaccess

Para configurar el .htaccess, tenemos que configurar el fichero apache2.conf y cambiar la directiva AllowOverride none por **All**

```
<Directory /var/www/>  
    Options Indexes FollowSymLinks  
    AllowOverride All  
    Require all granted  
</Directory>
```

# Instalar Servidor DNS

Lo primero que haremos será instalar el paquete bind9 con el comando: apt instar bind9.

Comprobamos que el servicio está activo:

```
miadmin@AOM-USED:~$ sudo service bind9 status
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-12-13 11:41:42 UTC; 46s ago
     Docs: man:named(8)
   Process: 2157 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 2158 (named)
    Tasks: 4 (limit: 2238)
   Memory: 7.0M
      CPU: 68ms
   CGroup: /system.slice/named.service
           └─2158 /usr/sbin/named -u bind

dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './DNSKEY/IN': 2001:500:2f::f#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './NS/IN': 2001:500:2f::f#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './DNSKEY/IN': 2001:500:12::d0d#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './NS/IN': 2001:500:12::d0d#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './DNSKEY/IN': 2001:500:2::c#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './NS/IN': 2001:500:2::c#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './DNSKEY/IN': 2001:503:c27::2:30#53
dic 13 11:41:42 AOM-USED named[2158]: network unreachable resolving './NS/IN': 2001:503:c27::2:30#53
dic 13 11:41:42 AOM-USED named[2158]: managed-keys-zone: Initializing automatic trust anchor management for 2
dic 13 11:41:42 AOM-USED named[2158]: resolver priming query complete: success
miadmin@AOM-USED:~$
```

Configuramos la red:

```
miadmin@AOM-USED: /etc/netplan
GNU nano 6.2
# This is the network config written by 'subiqu
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 192.168.3.214/24
      routes:
        - to: default
          via: 192.168.3.1
      nameservers:
        addresses: [192.168.3.214]
        search: [alejandro.local]
```

Editamos el fichero /etc/bind/named-conf.local

```
miadmin@AOM-USED: /etc/bind
GNU nano 6.2
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "alejandro.local"{
    type master;
    file "/etc/bind/db.alejandro.local";
};

zone "3.168.192.in-addr.arpa"{
    type master;
    file "/etc/bind/db.3.168.192.in-addr.arpa";
};
```

Comprobamos el fichero:

```
miadmin@AOM-USED:/etc/bind$ sudo nano named.conf.local
```

Creamos el fichero db.alejandro.local

```
miadmin@AOM-USED: /etc/bind
GNU nano 6.2
; BIND data for alejandro.local
;
$TTL      604800
@          IN SOA  AOM-USED.alejandro.local. root.localhost. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        3600       ; Negative Cache TTL
)
;
@          IN      NS    jnlldap.javier.local.
; Registros Host
@          IN      A      192.168.3.214
AOM-USED   IN      A      192.168.3.214
; Registros Alias
daw201     IN      CNAME  AOM-USED.alejandro.local.
daw202     IN      CNAME  AOM-USED.alejandro.local.
daw203     IN      CNAME  AOM-USED.alejandro.local.
daw204     IN      CNAME  AOM-USED.alejandro.local.
www.daw201 IN      CNAME  AOM-USED.alejandro.local.
www.daw202 IN      CNAME  AOM-USED.alejandro.local.
www.daw203 IN      CNAME  AOM-USED.alejandro.local.
www.daw204 IN      CNAME  AOM-USED.alejandro.local.
```

Comprobamos la zona:

```
miadmin@AOM-USED:/etc/bind$ sudo named-checkzone alejandro.local db.alejandro.local
zone alejandro.local/IN: loaded serial 1
OK
miadmin@AOM-USED:/etc/bind$
```

Para la zona inversa, copiamos el archivos de la zona directa.

```
miadmin@AOM-USED:/etc/bind$ sudo cp db.alejandro.local db.3.168.192.in-addr.arpa
```

Editamos el fichero nuevo y eliminar todos los registros CNAME.

```
miadmin@AOM-USED: /etc/bind
GNU nano 6.2
; BIND data for 3.168.192.in-addr.arpa
;
$TTL      604800
@          IN SOA      AOM-USED.alejandro.local. root.localhost. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                3600 )     ; Negative Cache TTL
;
@          IN      NS      AOM-USED.alejandro.local.
; Registros Host
; Registros Puntero
214        IN      PTR      alejandro.local.
214        IN      PTR      AOM-USED.alejandro.local.
```

Comprobamos la zona:

```
miadmin@AOM-USED: /etc/bind
miadmin@AOM-USED: /etc/bind$ sudo named-checkzone 3.168.192.in-addr.arpa db.3.168.192.in-addr.arpa
zone 3.168.192.in-addr.arpa/IN: loaded serial 1
OK
miadmin@AOM-USED: /etc/bind$
```

Reiniciamos el servicio

```
miadmin@AOM-USED: /etc/bind$ sudo service bind9 restart
```

Abrimos el puerto 53:

```
miadmin@AOM-USED: /etc/bind$ sudo ufw allow 53
Rule added
Rule added (v6)
miadmin@AOM-USED: /etc/bind$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
80 ALLOW Anywhere
9003 ALLOW Anywhere
8080/tcp ALLOW Anywhere
9003/tcp ALLOW Anywhere
9000 DENY Anywhere
81 ALLOW Anywhere
53 ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
9003 (v6) ALLOW Anywhere (v6)
8080/tcp (v6) ALLOW Anywhere (v6)
9003/tcp (v6) ALLOW Anywhere (v6)
9000 (v6) DENY Anywhere (v6)
81 (v6) ALLOW Anywhere (v6)
53 (v6) ALLOW Anywhere (v6)

miadmin@AOM-USED: /etc/bind$
```

Comprobamos la resolución de nombres con el comando NSLOOKUP:

```
miadmin@AOM-USED:/etc/bind$ nslookup
> alejandro.local
Server:      127.0.0.53
Address:     127.0.0.53#53

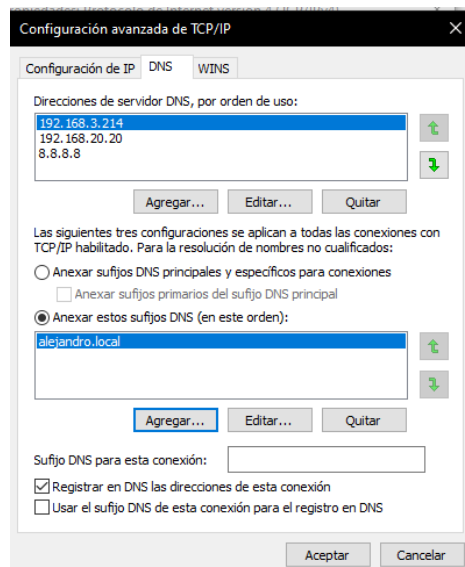
Non-authoritative answer:
Name:   alejandro.local
Address: 192.168.3.214
> AOM-USED
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   AOM-USED.alejandro.local
Address: 192.168.3.214
> daw201.alejandro.local
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
daw201.alejandro.local canonical name = AOM-USED.alejandro.local.
Name:   AOM-USED.alejandro.local
Address: 192.168.3.214
> daw201
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
daw201.alejandro.local canonical name = AOM-USED.alejandro.local.
Name:   AOM-USED.alejandro.local
Address: 192.168.3.214
>
```

Lo comprobamos en nuestra maquina Windows:



Lo comprobamos:

```
Microsoft Windows [Versión 10.0.19045.2251]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\daw2>nslookup
Servidor predeterminado: alejandro.local
Address: 192.168.3.214

> daw201
Servidor: alejandro.local
Address: 192.168.3.214

Nombre: AOM-USED.alejandro.local
Address: 192.168.3.214
Alias: daw201.alejandro.local

> AOM-USED
Servidor: alejandro.local
Address: 192.168.3.214

Nombre: AOM-USED.alejandro.local
Address: 192.168.3.214
```



# Instalación HTTPS

```
miadmin@AOM-USED: ~
```

```
miadmin@AOM-USED:~$ sudo openssl genrsa 2048 > daw202.key
```

```
miadmin@AOM-USED: ~
```

```
miadmin@AOM-USED:~$ sudo openssl req -new -key daw202.key > daw202.csr
```

```
miadmin@AOM-USED: ~
```

```
miadmin@AOM-USED:~$ sudo openssl req -new -key daw202.key > daw202.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Zamora
Locality Name (eg, city) []:Benavente
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Los Sauces
Organizational Unit Name (eg, section) []:Informatica y comunicaciones
Common Name (e.g. server FQDN or YOUR name) []:daw202.alejandro.local
Email Address []:alejandro.otamar@educa.jcyl.es
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
```

```
A challenge password []:
```

```
An optional company name []:
```

```
miadmin@AOM-USED:~$
```

```
miadmin@AOM-USED:~$ openssl x509 -req -days 365 -in daw202.csr -signkey daw202.key -out daw202.crt
Certificate request self-signature ok
subject=C = ES, ST = Zamora, L = Benavente, O = IES Los Sauces, OU = Informatica y comunicaciones, CN = daw202.alejandro.local, emailAddress = alejandro.otamar@educa.jcyl.es
miadmin@AOM-USED:~$
```

```
-in daw202.csr -signkey daw202.key -out daw202.crt
IES Los Sauces, OU = Informatica y comunicaciones, CN = daw202.alejandro.local, emailAddress = alejandro.otamar@educa.jcyl.es
```

Hacemos la misma operación con daw201.

Ahora, copiamos los archivos de .key

```
sudo cp daw202.key /etc/ssl/private
```

y lo mismo con daw201

Cambiamos el grupo con :

```
sudo chown root:ssl-cert /etc/ssl/private/daw202.key
```

cambiamos permisos:

```
sudo chmod 640 /etc/ssl/private/daw202.key
```

y lo mismo con daw201

copiamos los ficheros .crt

```
sudo cp /home/miadmin/daw202.crt /etc/ssl/certs/
```

y lo mismo con daw201

copiamos el archivo default-ssl.conf

```
sudo cp default-ssl.conf daw202.ssl.conf
```

y lo mismo con daw201

editamos los ficheros:

```
miadmin@AOM-USED: /etc/apache2/sites-available
miadmin@AOM-USED:/etc/apache2/sites-available$ cat daw202.ssl.conf
<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerAdmin webmaster@localhost
        ServerName daw202.alejandro.local
        DocumentRoot /var/www/daw202/public_html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/daw202-ssl-error.log
        CustomLog ${APACHE_LOG_DIR}/daw202-ssl-access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        #
        # SSL Engine Switch:
        # Enable/Disable SSL for this virtual host.
        SSLEngine on

        #
        # A self-signed (snakeoil) certificate can be created by installing
        # the ssl-cert package. See
        # /usr/share/doc/apache2/README.Debian.gz for more info.
        # If both key and certificate are stored in the same file, only the
        # SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/ssl/certs/daw202.crt
        SSLCertificateKeyFile   /etc/ssl/private/daw202.key
```

Lo mismo con daw201

activamos los sitios:

`sudo a2ensite daw202.ssl.conf`

y lo mismo con daw201

# Autenticación de acceso

## DIGEST

Lo primero, tenemos que instalar

```
miadmin@AOM-USED: ~  
miadmin@AOM-USED:~$ sudo a2enmod auth_digest  
[sudo] password for miadmin:  
Considering dependency authn_core for auth_digest:  
Module authn_core already enabled  
Module auth_digest already enabled  
miadmin@AOM-USED:~$
```

Seguidamente, crearemos una carpeta llamada DATA en /var/www/daw201/: (sudo mkdir data)

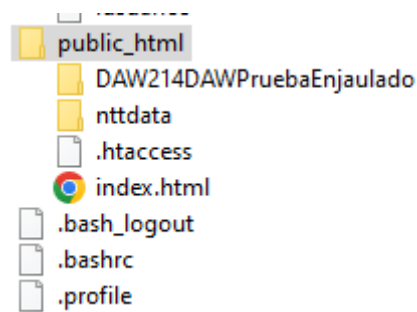
```
miadmin@AOM-USED: /var/www/daw201  
miadmin@AOM-USED:/var/www/daw201$ ll  
total 32  
dr-xr-xr-x 5 root root 4096 ene 19 09:39 ./  
drwxr-xr-x 6 root root 4096 ene 10 11:52 ../  
-rw-r--r-- 1 daw201 www-data 220 ene 6 2022 .bash_logout  
-rw-r--r-- 1 daw201 www-data 3771 ene 6 2022 .bashrc  
drwx----- 2 daw201 www-data 4096 oct 25 11:13 .cache/  
drwxr-xr-x 2 daw201 www-data 4096 ene 19 09:39 data/  
-rw-r--r-- 1 daw201 www-data 807 ene 6 2022 .profile  
drwxrwsr-x 3 daw201 www-data 4096 ene 17 11:59 public_html/  
miadmin@AOM-USED:/var/www/daw201$
```

Y su propietario, será daw201 en este caso y el grupo www-data (chown daw201 data/ ----- chgrp www-data data/)

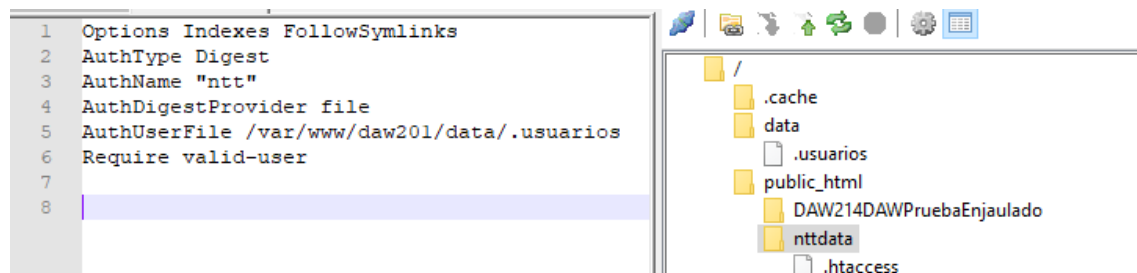
Acto seguido, crearemos dentro de la carpeta data/ un archivo y lo llamaremos: .usuarios y guardaremos los usuarios digest:

```
.usuarios  
1 jero:ntt:d1bcc00a46a995c217b4b5b734e89ce0  
2 israel:ntt:ab4dbb952c20585ea7537e65a673d616  
3 imelda:ntt:ee7f74e85b1b521bf1151af832019b7c
```

Ahora, crearemos el directorio nttdata en la carpeta public\_html/



Dentro de esta carpeta, crearemos un archivo .htaccess y le añadiremos las siguientes directivas y grabamos:



Require valid-user(Hace que pueda acceder cualquier usuario guardado en el fichero .usuarios)

Para hacerlo por grupos, crearemos el fichero .grupos en la carpeta /data

Activaremos el modulo authz\_groupfile y reiniciamos el servicio apache.

```
miadmin@AOM-USED: /var/www/daw201/data
miadmin@AOM-USED:/var/www/daw201/data$ sudo a2enmod authz_groupfile
[sudo] password for miadmin:
Considering dependency authz_core for authz_groupfile:
Module authz_core already enabled
Enabling module authz_groupfile.
To activate the new configuration, you need to run:
    systemctl restart apache2
miadmin@AOM-USED:/var/www/daw201/data$
```

En el fichero .htaccess añadiremos las directivas:

```
6 AuthGroupFile /var/www/daw201/data/.grupos
7 Require group jefeproyectos
```

Y para los grupos quitamos la de: Require valid-user