

# TALLER DE LINEA DE COMANDOS LINUX

Instalación de Java 11, Tomcat 9, Python 3, Gunicorn, Nginx, MySQL 8

Despliegue de aplicaciones en Spring Boot 2.5.5, Django, Vue 3

Laura Pulido Díaz - Octubre de 2021

## Definición de Alcance

### Objetivos del taller:

1. El estudiante participará activamente en el reconocimiento de entornos Linux por medio de línea de comandos desde el conocimiento general que debe tener un desarrollador, no específicos de un administrador de sistemas y plataformas.
2. El estudiante comprenderá los conceptos generales asociados a un servidor físico, servidor virtual, servidor de base de datos y servidor de aplicaciones.
3. El estudiante implementará servidores de aplicaciones y bases de datos requeridos sobre entorno Linux para el despliegue de sus aplicativos.
4. El estudiante realizará la carga directa a los servidores de aplicaciones dispuestos de sus aplicativos para las pruebas necesarias en el servidor.

### Resumen - Alcance:

Este documento describe un taller de instalación y configuración de servidores en servidor virtual Oracle Linux en OCI (Oracle Cloud Infraestructure -> Nube de Oracle), distribución basada en RedHat, para montaje de aplicaciones Backend (Servicios Rest): Django, SpringBoot y Frontend: VueJS (Vue3) para el despliegue de un aplicativo SPA. Incluye uso de herramientas básicas y comandos de Linux para navegación y gestión de archivos y directorios.

### Futuros – Limitaciones de Alcance:

La configuración presentada en este documento no presenta configuración de contenedores o despliegue por microservicios, ni pipelines para integración continua para realizar despliegues profesionales de aplicativos en las tecnologías listadas.

El uso del repositorio no se está aprovechando con una estrategia adecuada para la gestión de ramas que facilite el mantenimiento del aplicativo por parte de un equipo de desarrollo, ninguno de los repositorios se ha integrado a las herramientas de desarrollo ni a las de despliegue, por lo que GitHub (herramienta utilizada para gestionar repositorios en la nube) no se incluye en este documento.

No se utilizaron diferentes ambientes para producción, QA y desarrollo.

No se incluyó configuración de certificados de seguridad para comunicación https entre los diferentes servidores de aplicaciones ni hacia el cliente.

\*\*\* importante\*\*\* No se incluyó conceptualización de DevOps para entender las responsabilidades de un desarrollador en sus despliegues sobre un servidor.

## Información previa del curso:

[https://unabedu-my.sharepoint.com/:f/g/personal/laura\\_pulido\\_o365\\_unab\\_edu\\_co/EoCfux7K6kBEmlJEkmAS00BEGUiorNacaKo4RNCLmfig](https://unabedu-my.sharepoint.com/:f/g/personal/laura_pulido_o365_unab_edu_co/EoCfux7K6kBEmlJEkmAS00BEGUiorNacaKo4RNCLmfig)

## Preparación

### 1. Herramientas para Acceso al Servidor desde Windows

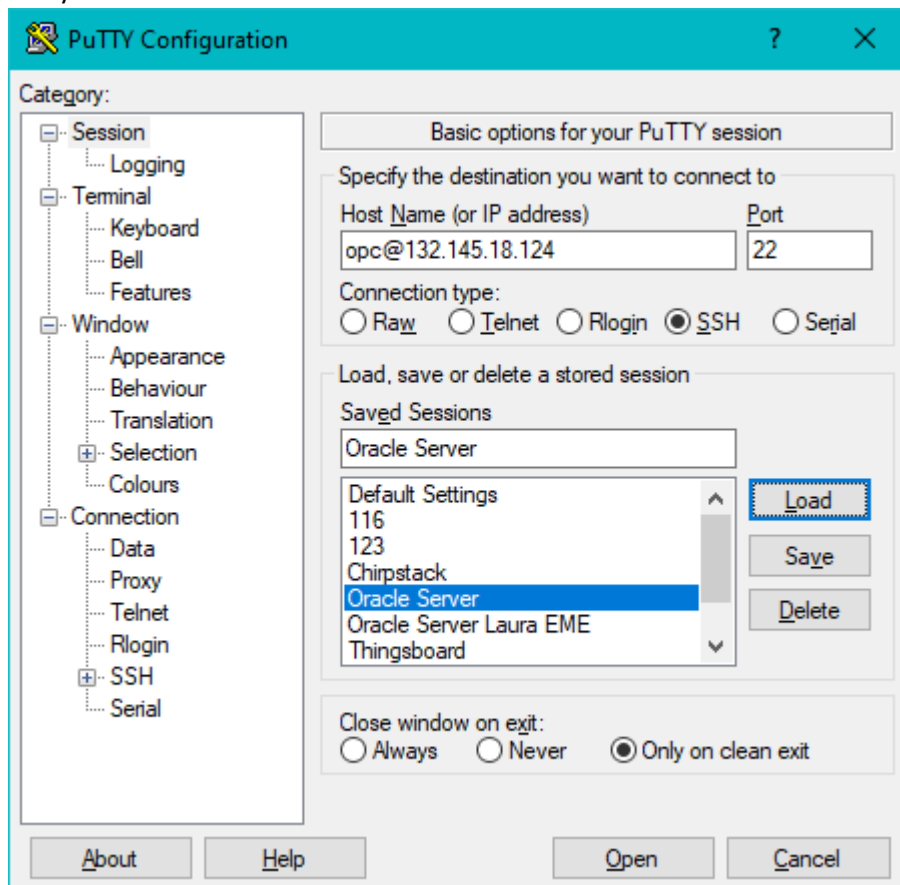
- WinSCP: <https://winscp.net/eng/download.php>
- Putty: <https://www.putty.org/>  
<https://www.hostinger.co/tutoriales/conectar-usando-terminal-putty-ssh/>

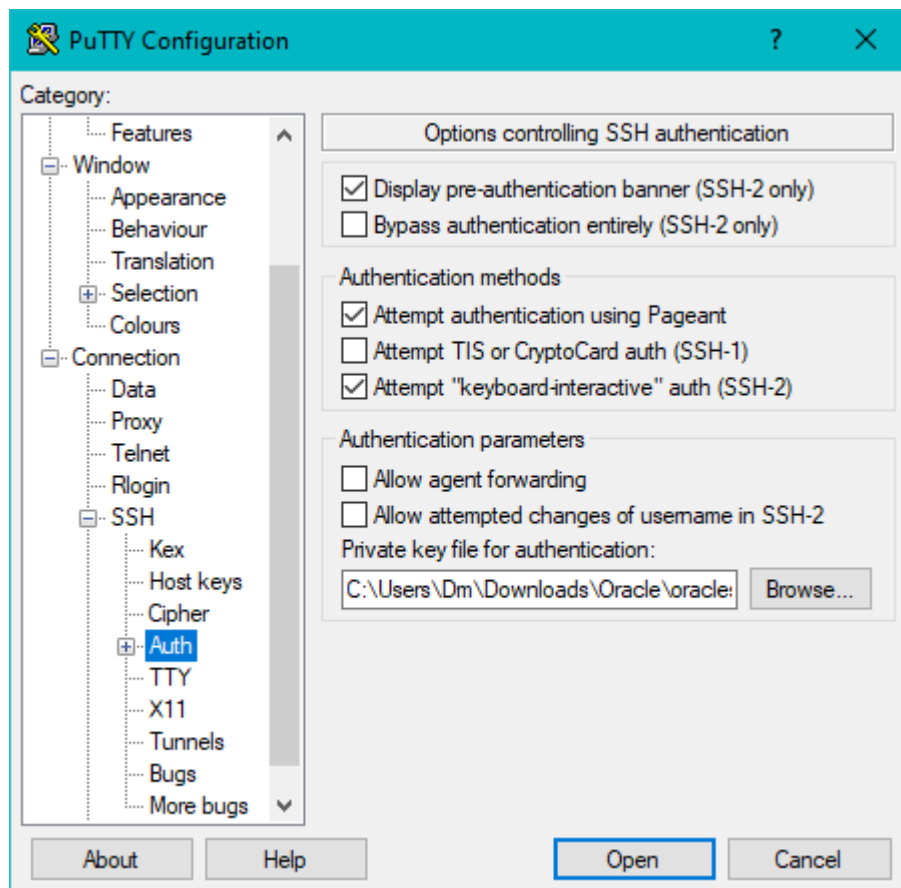
### 2. Creación de Llaves PPK

Remitirse al documento disponible en línea: Tutoriales/Procedimiento para Conexión SSH a VM Linux.pdf

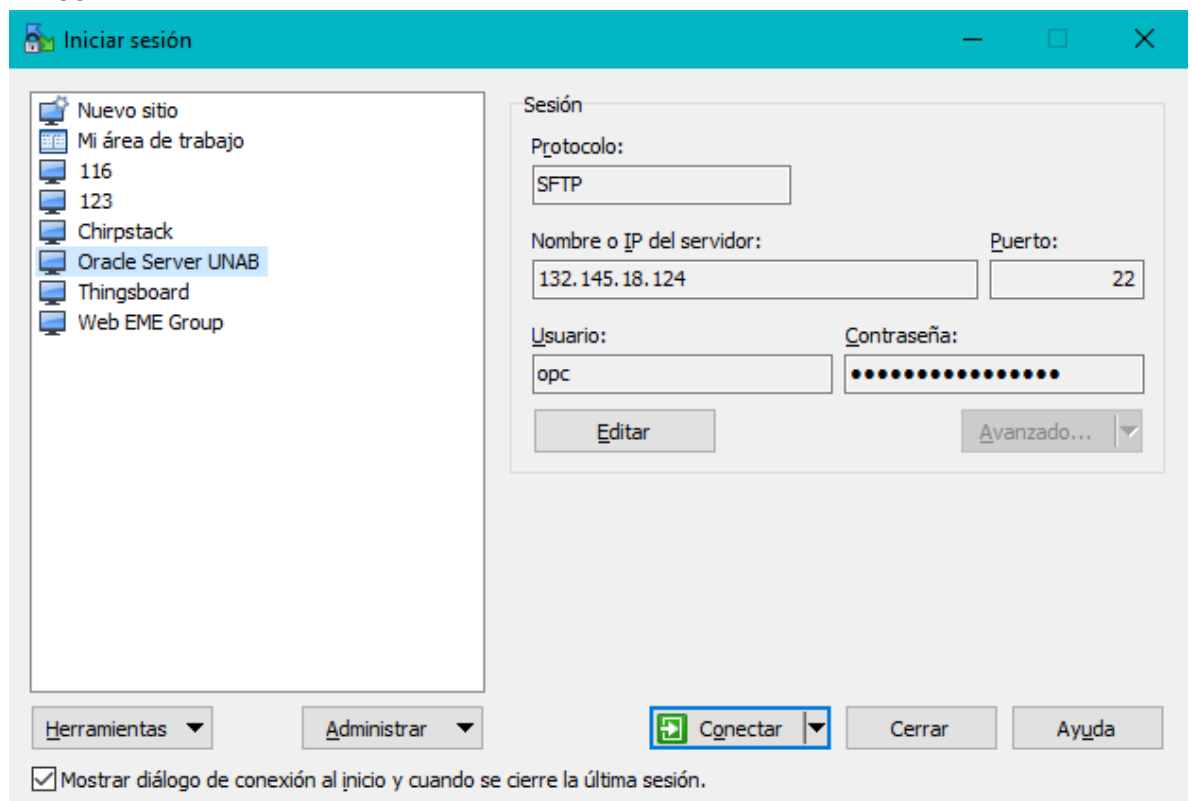
### 3. Configuración de conexión con llaves privadas:

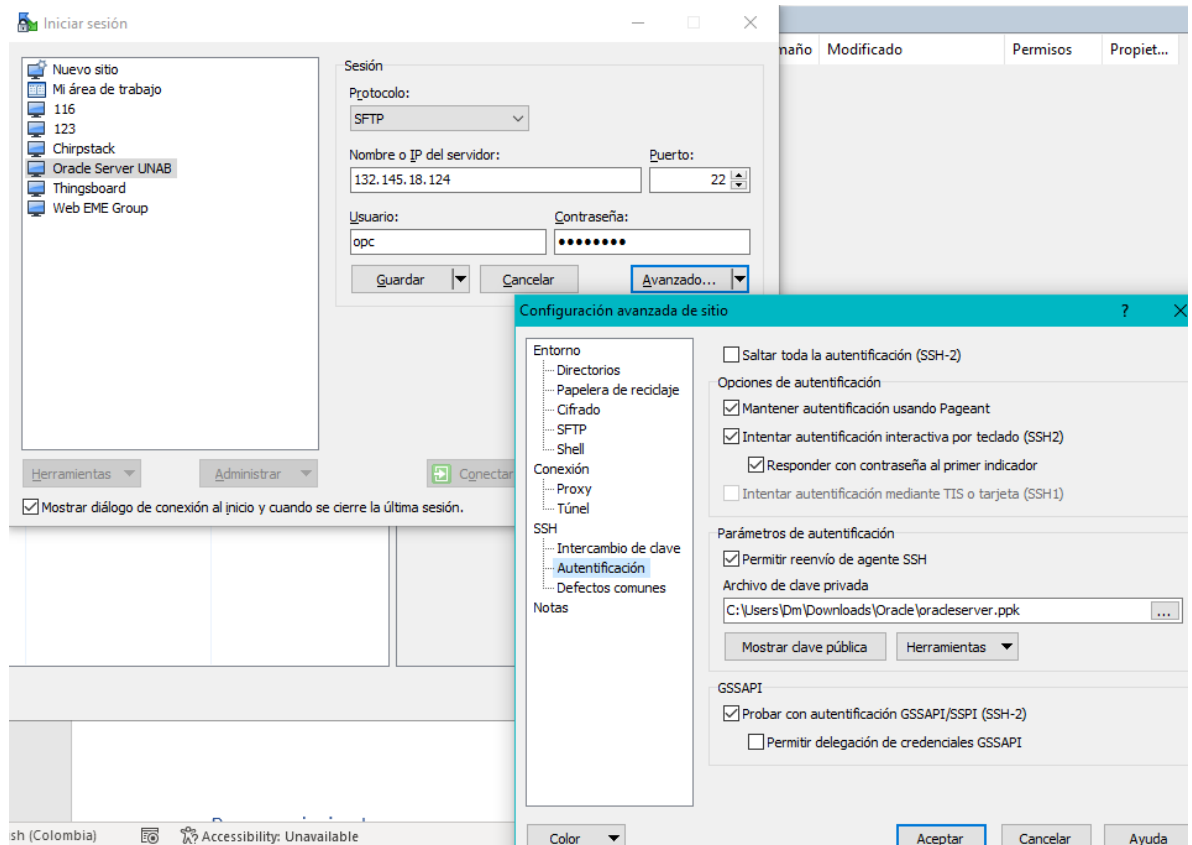
- Putty:





- WinSCP:





## Reconocimiento

Revisar uno a uno los siguientes comandos de Linux, empezando con el comando `man`, permitiendo que este sea de ayuda para entender todos los demás comandos disponibles en el listado.

4. `man` -> muestra el manual de cada comando para entender cómo se usa
5. `history` -> muestra el historial de comandos que se han ejecutado
6. Información del sistema operativo y la máquina
  - `hostname`: nombre del usuario del sistema
  - `uname -n -v -r -m -a`
  - `lshw -short`
  - `lsblk -a`
  - `sudo fdisk -l`
  - `free -mt`
  - `top`
7. Navegación entre carpetas
  - `pwd`
  - `cd ..` (con dos puntos) para ir un directorio hacia arriba

- cd para ir directamente a la carpeta de inicio
- cd- (con un guión) para ir al directorio anterior
- ls -R también listará todos los archivos en los subdirectorios
- ls -a mostrará los archivos ocultos
- ls -al listará los archivos y directorios con información detallada como los permisos, el tamaño, el propietario, etc.

#### 8. Crear, copiar, mover y eliminar archivos y carpetas

- cat > nombreadarchivo crea un nuevo archivo.
- cat nombreadarchivo1 nombreadarchivo2>nombreadarchivo3 une dos archivos (1 y 2) y almacena la salida de ellos en un nuevo archivo (3)
- convertir un archivo a mayúsculas o minúsculas, cat nombreadarchivo | tr a-z A-Z> salida.txt
- cp para copiar archivos del directorio actual a un directorio diferente
- mv es mover archivos, aunque también se puede usar para cambiar el nombre de los archivos
- mkdir para crear un nuevo directorio
- rmdir te permite eliminar directorios vacíos
- rm se usa para eliminar directorios y el contenido dentro de ellos
- touch te permite crear un nuevo archivo en blanco
- locate -i escuela\*nota buscará cualquier archivo que contenga la palabra «escuela» y «nota», ya sea en mayúsculas o minúsculas
- el comando find /home/ -name notas.txt buscará un archivo llamado notas.txt dentro del directorio de inicio y sus subdirectorios
- grep azul notepad.txt buscará la palabra azul en el archivo
- tar es el comando más utilizado para guardar múltiples archivos en un tarball, un formato de archivo de Linux común que es similar al formato zip
- wget descargar archivos de Internet
- echo Hola, mi nombre es John >> nombre.txt para escribir texto en archivo
- zip, unzip
- 

#### 9. Permisos y usuarios

- sudo «SuperUser Do» (SuperUsuario hace)
- chmod <https://www.computerhope.com/unix/uchmod.htm>
- chown
- useradd, userdel

#### 10. Comandos de servicios y yum

- sudo systemctl list-unit-files --type service --all
- sudo systemctl | grep running
- sudo systemctl start/stop/status/enable/disable [nombre\_servicio]
- sudo netstat -plnt
- yum [https://www.linuxtotal.com.mx/index.php?cont=info\\_admon\\_020](https://www.linuxtotal.com.mx/index.php?cont=info_admon_020)

11. Editor VI : <https://docs.oracle.com/cd/E19620-01/805-7644/6j76klopr/index.html>

## 12. Estructura de carpetas de Linux



bin	Binarios de usuario
boot	Ejecutables y archivos requeridos para el arranque
dev	Archivos de información de todos los volúmenes
etc	Archivos de configuración del sistema y de aplicaciones
home	Directorio personal con las carpetas de usuario
lib	Bibliotecas necesarias para la ejecución de binarios
media	Directorio de montaje de volúmenes extraíbles
opt	Ficheros de aplicaciones externas que no se integran en /usr
proc	Ficheros de información de procesos
root	Directorio personal de superusuario
sbin	Binarios de sistema
srv	Archivos relativos a servidores web, FTP, etc.
sys	Archivos virtuales con información de eventos del sistema
tmp	Directorio de ficheros temporales
usr	Archivos de programas y aplicaciones instaladas
var	Archivos de variables, logs, emails de los usuarios del sistema, etc.

## Ejercicio

- Utilizando únicamente la consola, crear un directorio llamado "animales" dentro del directorio home del usuario y dentro de este crear 3 archivos llamados "gato", "perro" y "pez"
- Usando el comando echo escribir en el archivo pez: "Esto es un archivo de pececitos".
- Usando el editor VI escribir en el archivo llamado perro "El perro es el mejor amigo del hombre y existen muchas razas diferentes".
- Utilizando comandos de búsqueda buscar todos los archivos que contengan "pe" en el nombre.

## Instalación

### 13. Limpiar el servidor

- `yum list installed > list_packs.txt`
- `sudo yum remove jboss*`
- `sudo yum remove maven*`
- `sudo yum remove mysql*`
- `sudo yum remove nginx*`
- `sudo yum remove postgres*`
- `sudo yum remove python3`
- `sudo yum remove tomcat`
- `sudo yum remove tomcat*`
- `sudo yum remove apache*`
- `yum list installed > list_packs_post.txt`
- `sudo yum upgrade -y`
- `sudo reboot`
- `sudo yum groupinstall "Development Tools"`

### 14. Java 11

- `java -version`
- `wget --no-check-certificate -c --header "Cookie: oraclelicense=accept-securebackup-cookie" https://download.oracle.com/otn-pub/java/jdk/11.0.12%2B8/f411702ca7704a54a79ead0c2e0942a3/jdk-11.0.12_linux-aarch64_bin.rpm`
- `sudo rpm -iv jdk-11.0.12_linux-aarch64_bin.rpm`
- `sudo alternatives --config java`
- `java -version`

### 15. Python 3

- `python --version`
- `sudo yum install python3 python3-pip python3-devel`
- `python3 --version`
- `sudo pip3 install -U pip requests`
- `python3 -m venv python3-virtualenv`
- `source python3-virtualenv/bin/activate`
- `python --version`
- `pip install -U pip requests`
- `pip freeze`
- `deactivate`

## 16. Mysql 8

- `sudo rpm -Uvh https://repo.mysql.com/mysql80-community-release-el7-3.noarch.rpm`
- `sudo sed -i 's/enabled=1/enabled=0/' /etc/yum.repos.d/mysql-community.repo`
- `sudo yum --enablerepo=mysql80-community install mysql-community-server`
- `sudo systemctl enable mysqld`
- `sudo systemctl start mysqld`
- `sudo grep "A temporary password" /var/log/mysqld.log`
- `sudo mysql_secure_installation`
- `service mysqld restart`
- `sudo service mysqld restart`
- `sudo chkconfig mysqld on`
- `mysql -h localhost -u root -p`
- `exit`

## 17. Tomcat 9

- `sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat`
- `wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.54/bin/apache-tomcat-9.0.54.zip -P /tmp`
- `sudo mkdir -p /opt/tomcat`
- `cd tmp`
- `cd /tmp`
- `unzip apache-tomcat-9.0.54.zip`
- `mv apache-tomcat-9.0.54 /opt/tomcat/`
- `sudo mv apache-tomcat-9.0.54 /opt/tomcat/`
- `sudo ln -s /opt/tomcat/apache-tomcat-9.0.54 /opt/tomcat/latest`
- `sudo chown -R tomcat: /opt/tomcat`
- `sudo sh -c 'chmod +x /opt/tomcat/latest/bin/*.sh'`
- `sudo nano /etc/systemd/system/tomcat.service`
- `sudo systemctl daemon-reload`
- `sudo systemctl start tomcat`
- `sudo systemctl status tomcat`
- `sudo systemctl enable tomcat`
- `sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp`
- `sudo firewall-cmd --reload`
- `sudo vi /opt/tomcat/latest/conf/tomcat-users.xml`
- `*** AQUÍ SE DEBEN ACTUALIZAR USUARIOS Y ROLES`
  - `<role rolename="admin-gui"/>`
  - `<role rolename="manager-gui"/>`
  - `<user username="nombreusuario" password="claveusuario" roles="admin-gui,manager-gui"/>`
- `sudo systemctl restart tomcat`

## 18. Gunicorn

- `source python3-virtualenv/bin/activate`
- `pip install gunicorn`
- `sudo yum update`



- sudo yum upgrade -y
- sudo yum groupinstall "Development Tools"
- sudo yum install build-essential checkinstall
- deactivate

## 19. Nginx

- sudo yum-config-manager --enable \\*
- sudo yum-config-manager --disable mysql\\*
- sudo yum-config-manager --disable ol7\_optional\_archive
- sudo yum repolist all
- sudo yum install -y nginx
- sudo systemctl enable --now nginx.service
- sudo systemctl status nginx
- sudo firewall-cmd --add-service=http --permanent
- sudo firewall-cmd --reload

## Despliegue

### 20. Herramientas para los siguientes pasos:

- Eclipse o Spring Studio Tools + Proyecto Java Spring Boot
- Pycharm + Proyecto Django
- Visual Studio Code + Proyecto Vue
- Postman + Abrir workspace con las consultas para probar Backend. Duplicar consultas apuntando a la IP del servidor asignado
- Paciencia, conexión a internet, Discord, archivo de Word para anotar cualquier error que se genere Google y Stack Overflow

### 21. Crear BD

- mysql -h localhost -u root -p
- \*\*\* AQUÍ PIDE EL PASSWORD DE SU USUARIO ROOT DE LA BASE DE DATOS \*\*\*
- CREATE DATABASE nombre\_de\_la\_base\_de\_datos;
- CREATE USER 'nombre\_usuario'@'localhost' IDENTIFIED BY 'clave\_del\_usuario';
- GRANT ALL PRIVILEGES ON nombre\_de\_la\_base\_de\_datos.\* TO 'nombre\_usuario'@'localhost';
- exit

### 22. Desplegar Backend Spring Boot

\*\*\* Realizar ajustes a la configuración de base de datos, nos aseguramos que el usuario, clave y nombre de la base sean los mismos que creamos en el servidor de mysql. Tener en cuenta que en el servidor se exigió una clave segura.

Modificamos el proyecto, para despliegues en servidor externo:

1. La clase principal debe extender de ServletInitializer:
  - extends SpringBootServletInitializer

## 2. Modificaciones al archivo pom.xml

- Después de etiqueta name en la parte superior del archivo, agregar etiqueta packaging:  
`<packaging>war</packaging>`
- Modificar la dependencia de spring boot web starter vamos a excluir el servidor embebido:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

- Agregamos otra dependencia para tomcat como servidor externo, la ubicamos antes de finalizar la etiqueta dependencies:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-tomcat</artifactId>
  <scope>provided</scope>
</dependency>
```

- En la etiqueta build, al final del archivo, agregamos un nombre para el archivo empaquetado WAR

```
<build>
  <finalName>${project.artifactId}</finalName>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

## 3. Ejecutamos:

- Click derecho
- Run as
- Maven Install

## 4. Localizamos el archivo WAR en la carpeta del proyecto/target

## 5. Desplegamos el WAR en Tomcat 9:

- Entrar en un navegador a localhost:8080/manager/
- Ingresar las credenciales configuradas en el archivo de tomcat users.
- Debajo de la lista de aplicaciones, seleccionar el archivo proyecto.war y dar click en Desplegar:

Archivo WAR a desplegar	
Seleccione archivo WAR a cargar	<input type="button" value="Seleccionar archivo"/> demo.war
<input type="button" value="Desplegar"/>	

## 23. Desplegar Backend Django

\*\*\* Realizar ajustes a la configuración de base de datos, nos aseguramos de que el usuario, clave y nombre de la base sean los mismos que creamos en el servidor de mysql. Tener en cuenta que en el servidor se exigió una clave segura.

- source python3-virtualenv/bin/activate
- cd /home/opc
- mkdir Proyecto
- *\*\*\* Cargar el código del proyecto a esta carpeta en el servidor. Se puede usar WinSCP.*
- cd Proyecto
- pip install -r requirements.txt
- python manage.py migrate
- python manage.py createsuperuser
- python manage.py runserver
- *\*\*\* El comando anterior debe ejecutar un servidor embebido*
- CTRL + C
- gunicorn --bind 0.0.0.0:8000 nombreproyecto.wsgi
- *\*\*\* El comando anterior debe ejecutar un servidor GUnicorn, validar que no haya problemas para la ejecución*
- CTRL + C
- deactivate
- sudo vi /etc/systemd/system/gunicorn.socket
- INSERT

*Escribir esto en el archivo:*

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
```

- ESC
- :wq
- sudo vi /etc/systemd/system/gunicorn.service
- INSERT

*Escribir esto en el archivo:*

```
[Unit]
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
User=opc
Group=www-data
WorkingDirectory=/home/opc/Proyecto
ExecStart=/home/opc/python3-virtualenv/bin/gunicorn \
    --access-logfile - \
    --workers 3 \
```

```
--bind unix:/run/gunicorn.sock \  
Proyecto.wsgi:application
```

```
[Install]  
WantedBy=multi-user.target
```

- ESC
- :wq
- sudo systemctl restart gunicorn.socket
- sudo systemctl enable gunicorn.socket
- sudo systemctl daemon-reload
- sudo systemctl restart gunicorn
- sudo touch /etc/nginx/conf.d/default.conf
- sudo chmod 777 /etc/nginx/conf.d/default.conf
- INSERT

Escribir esto en el archivo:

```
server {  
    listen 81;  
    server_name 132.145.18.126;  
    location /static/{  
        root /home/opc/Proyecto;  
    }  
  
    location / {  
        include proxy_params;  
        proxy_pass http://unix:/run/gunicorn.sock;  
    }  
}  
EOF
```

- ESC
- :wq
- sudo systemctl restart nginx.service

## 24. Desplegar Frontend Vue

\*\*\* Realizar ajustes de proyecto: apuntar a la IP del servidor y si se requiere un puerto (si es 80 no es necesario poner el puerto) en lugar de localhost en cada petición (fetch).

En la terminal de visual studio, ejecutar el comando:

- npm run build

Este comando va a compilar el proyecto creando una carpeta **"dist"**, esta carpeta contiene todos los archivos que se deben subir al servidor.

Crear un directorio y configurarlo en el servidor NGINX para el proyecto:

- sudo mkdir /srv/vuefront
- sudo chown -R nginx:nginx /srv/vuefront

- `sudo chcon -Rt httpd_sys_content_t /srv/vuefront`
- `vi /etc/nginx/conf.d/default.conf`
- INSERT

*Escribir esto en el archivo:*

```
server {  
    listen    80;  
    server_name 132.145.18.126;  
    root      /srv/vuefront;  
    index     index.html;  
}  
EOF
```

- ESC
- :wq
- `sudo chmod 777 -R /srv/vuefront/`
- `sudo systemctl restart nginx.service`

*Entrar a la carpeta /srv/vuefront/ en el servidor, por WinSCP o cualquier herramienta de SFTP y cargar el contenido completo de la carpeta "dist" del proyecto que se creó dentro de la carpeta donde está el código fuente del proyecto incluyendo archivos y directorios.*