



**UNIVERSIDAD ORT**  
**Uruguay**

**OBLIGATORIO TALLER DE  
SERVIDORES LINUX**

Leonardo Martínez - Estudiante Nº 208194  
Nicolás Marrero - Estudiante Nº 191020

CONTENIDO	
INTRODUCCIÓN	4
Propuesta	4
SERVIDORES	5
Controlador	5
Especificaciones	5
Ubuntu	6
Especificaciones	6
LVM y Particionado	7
Centos	7
Especificaciones	7
LVM y Particionado	8
GIT	9
Playbooks	10
PlaybookCentos	10
Instalar JDK de Java	10
Crear usuario Tomcat	10
Crear grupo Tomcat	10
Descargar Tomcat	10
Crear directorio de instalación de Tomcat	10
Instalar unzip	11
Instalar tar	11
Extraer Tomcat	11
Cambiar permisos del directorio de Tomcat	11
Crear servicio systemd para Tomcat:	11
Recargar systemd	11
Habilitar e iniciar Tomcat	12
Desplegar el archivo WAR	12
Copiar el archivo de configuración de la aplicación	12
Habilitar puertos en el firewall	12
Recargar el firewall	12
PlaybookUbuntu	13

Configurar y asegurar MariaDB en Ubuntu	13
Configurar el intérprete de Python	13
Actualizar la lista de paquetes	13
Instalar MariaDB	13
Instalar MariaDB y herramientas relacionadas	13
Verificar si mysql_secure_installation está instalado	14
Mostrar ruta de mysql_secure_installation	14
Ejecutar mysql_secure_installation	15
BIBLIOGRAFÍA Y REFERENCIAS	16



## INTRODUCCIÓN

El presente documento describe el procedimiento para la realización del obligatorio del taller de redes de la carrera Analista en Infraestructura Informática.

Para la realización de este obligatorio se utilizaron las siguientes herramientas:

- VirtualBox como software de virtualización
- Cockpit como medio de acceso a todos los servidores virtualizados
- Git para documentar, gestionar y respaldar los archivos/documentos
- Ansible para la ejecución de los automatismos.

## PROPUESTA

La letra propone la implementación de 3 servidores Linux.

- 1 servidor que oficiará como controlador de ansible
- 1 servidor que oficiará como servidor de aplicaciones
- 1 servidor que oficiará como servidor de base de datos

Todo el proceso de gestión se realizará mediante playbooks de ansible que se encargarán de realizar las instalaciones y/o configuraciones en los servidores pertinentes.

## SERVIDORES

### CONTROLADOR

El controlador es el servidor encargado de alojar el servicio de ansible, con sus respectivos playbooks, inventarios y dependencias; cuya función será desplegar las aplicaciones, servicios y configuraciones en su servidor-destino correspondiente.

En este entorno el controlador también cuenta con apache configurado con un virtual host que realiza un ProxyPass al servicio de cockpit el cual se utilizó durante todo el proceso como herramienta de conexión remota a todos los servidores.

### ESPECIFICACIONES

- Tipo de máquina: Máquina Virtual
- Procesadores: 1 núcleo
- Memoria: 2GB
- Almacenamiento físico: 25GB full disk 1 partición
- SO: Ubuntu 24.04 LTS

```
user@controlador ~> lscpu | grep CPU && free -m
CPU op-mode(s): 32-bit, 64-bit
CPU(s): 1
On-line CPU(s) list: 0
CPU family: 6
NUMA node0 CPU(s): 0
Vulnerability Mmio stale data: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
total used free shared buff/cache available
Mem: 1968 549 411 1 1197 1418
Swap: 2043 0 2043
user@controlador ~>
```

```
user@controlador ~> cat /etc/os-release | grep NAME
PRETTY_NAME="Ubuntu 24.04 LTS"
NAME="Ubuntu"
VERSION_CODENAME=noble
UBUNTU_CODENAME=noble
```

## UBUNTU

Este servidor alojará el servidor de base de datos a través del servicio MySQL para las distintas aplicaciones que lo requieran. Para este caso la aplicación que lo consumirá será “ToDo” alojada en el servidor Centos.

## ESPECIFICACIONES

- Tipo de máquina: Máquina Virtual
- Procesadores: 1 núcleo
- Memoria: 2GB
- Almacenamiento físico: 13GB particionado con LVM según los requerimientos
- SO: Ubuntu 24.04 LTS

```
user@ubuntu:~$ cat /etc/os-release | grep NAME
PRETTY_NAME="Ubuntu 24.04 LTS"
NAME="Ubuntu"
VERSION_CODENAME=noble
UBUNTU_CODENAME=noble
```

```
user@ubuntu ~-> lscpu | grep CPU && free -m
CPU op-mode(s): 32-bit, 64-bit
CPU(s): 1
On-line CPU(s) list: 0
CPU family: 6
NUMA node0 CPU(s): 0
Vulnerability Mmio stale data: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
Mem: total used free shared buff/cache available
Swap: 2047 0 2047
```

## LVM Y PARTICIONADO

Se particionó el disco duro durante la instalación en base a los requerimientos de la siguiente forma:

- Partición de 1GB para /boot
- LVM de 7GB para /
- LVM de 3GB para /var
- LVM de 2GB para SWAP

### Storage configuration

#### RESUMEN DEL SISTEMA DE ARCHIVOS

PUNTO DE MONTAJE	TAMAÑO	TIPO	TIPO DE DISPOSITIVO	
[ /	7.000G	new ext4	new LVM logical volume	▶ ]
[ /boot	1.000G	new ext4	new partition of disco local	▶ ]
[ /var	3.000G	new ext4	new LVM logical volume	▶ ]
[ SWAP	1.996G	new swap	new LVM logical volume	▶ ]

## CENTOS

Este servidor alojará el servidor de aplicaciones a través del servicio Tomcat. Para este entorno la aplicación desplegada será "ToDo" provista por el docente.

## ESPECIFICACIONES

- Tipo de máquina: Máquina Virtual
- Procesadores: 1 núcleo
- Memoria: 2GB
- Almacenamiento físico: 13GB particionado con LVM según los requerimientos
- SO: Centos Stream 9

```
user@centos ~> lscpu | grep CPU && free -m
modo(s) de operación de las CPUs: 32-bit, 64-bit
CPU(s): 1
Lista de la(s) CPU(s) en línea: 0
Familia de CPU: 6
CPU(s) del nodo NUMA 0: 0
Vulnerabilidad Mmio stale data: Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
total      used      free     shared  buff/cache   available
Mem:      1774        453       1090         4         377        1321
Swap:      2039          0       2039
```

```
user@centos ~> cat /etc/os-release | grep NAME
NAME="CentOS Stream"
PRETTY_NAME="CentOS Stream 9"
CPE_NAME="cpe:/o:centos:centos:9"
```

## LVM Y PARTICIONADO

Se particionó el disco duro durante la instalación en base a los requerimientos de la siguiente forma:

- Partición de 1GB para /boot
- LVM de 7GB para /
- LVM de 3GB para /var
- LVM de 2GB para SWAP

Nueva instalación CentOS Stream 9

SISTEMA

/

cs-root

7 GiB >

/var

cs-var

3 GiB

/boot

sda1

1024 MiB

swap

cs-swap

1,99 GiB

RESUMEN DE CAMBIOS

Sus personalizaciones harán que los siguientes cambios tengan efecto tras volver al menú principal y comenzar la instalación:

Orden	Acción	Tipo	Dispositivo	Punto de Montaje
1	destruir formato	Unknown	ATA VBOX HARDDISK (sda)	
2	crear formato	Tabla de particiones (MSDOS)	ATA VBOX HARDDISK (sda)	
3	crear dispositivo	partition	sda1 en ATA VBOX HARDDISK	
4	crear dispositivo	partition	sda2 en ATA VBOX HARDDISK	
5	crear formato	physical volume (LVM)	sda2 en ATA VBOX HARDDISK	
6	crear dispositivo	lvmvg	cs	
7	crear dispositivo	lvmlv	cs-var	
8	crear formato	xfs	cs-var	/var
9	crear dispositivo	lvmlv	cs-root	
10	crear formato	xfs	cs-root	/
11	crear dispositivo	lvmlv	cs-swap	
12	crear formato	swap	cs-swap	
13	crear formato	xfs	sda1 en ATA VBOX HARDDISK	/boot

Cancelar y Volver al particionado personalizado

Aceptar cambios

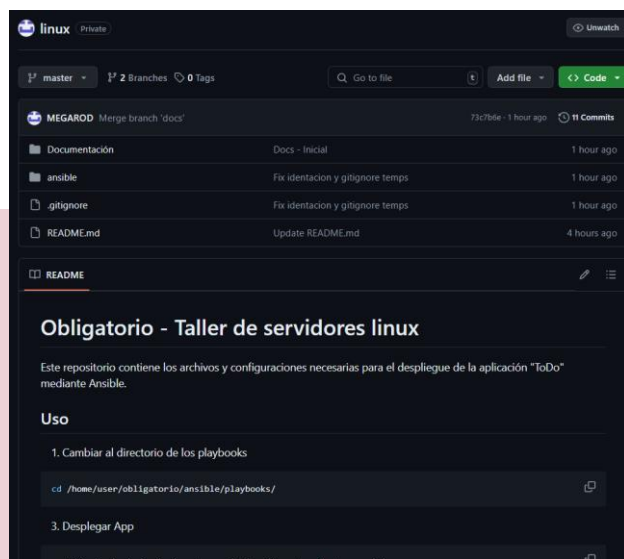
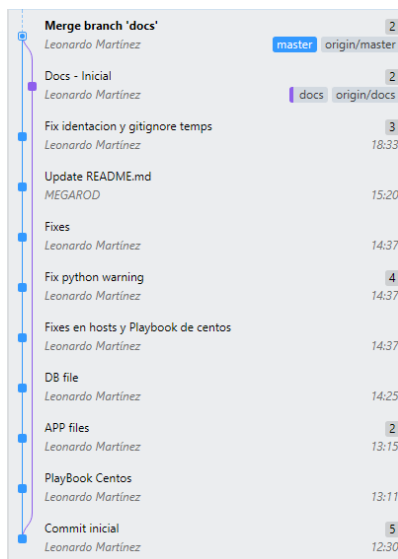


## GIT

Se utilizó la herramienta GIT para realizar la gestión de código, versionado y backups de las configuraciones de los playbooks, inventarios de hosts, archivos de dependencias y documentación.

Se inicializó el repositorio local de la carpeta `/home/user/obligatorio/` del controlador de ansible y se vinculó con el repositorio remoto `github.com/MEGAROD/linux`

La documentación del obligatorio se realizó en una Branch distinta llamada “docs”



## PLAYBOOKS

### PLAYBOOKCENTOS

---

#### INSTALAR JDK DE JAVA

Esta tarea instala el Java Development Kit (JDK) en el sistema. El JDK es un conjunto de herramientas necesarias para desarrollar aplicaciones Java, que incluye el compilador y las bibliotecas necesarias. La tarea asegura que el JDK esté disponible y actualizado en el sistema.

---

#### CREAR USUARIO TOMCAT

Esta tarea crea un nuevo usuario en el sistema llamado "Tomcat". Este usuario es utilizado para ejecutar el servidor de aplicaciones Tomcat de forma segura, separando los permisos y accesos de Tomcat de los de otros usuarios del sistema. Esto ayuda a mejorar la seguridad y el control del entorno de ejecución de Tomcat.

---

#### CREAR GRUPO TOMCAT

Esta tarea crea un nuevo grupo en el sistema llamado "Tomcat". Este grupo puede ser utilizado para gestionar los permisos y accesos de los usuarios relacionados con el servidor de aplicaciones Tomcat, facilitando la administración y seguridad al agrupar a los usuarios que necesitan acceso a los mismos recursos o archivos asociados con Tomcat.

---

#### DESCARGAR TOMCAT

Esta tarea descarga el archivo comprimido de Tomcat desde una fuente específica (generalmente un sitio web oficial o un repositorio). La descarga de Tomcat es el primer paso para instalar y configurar el servidor de aplicaciones en el sistema, permitiendo que posteriormente se extraiga y se configure para su uso.

---

#### CREAR DIRECTORIO DE INSTALACIÓN DE TOMCAT

Esta tarea crea un directorio específico en el sistema donde se instalará Tomcat. Este directorio sirve como la ubicación base para almacenar los archivos del servidor de aplicaciones Tomcat, incluyendo su configuración, bibliotecas y aplicaciones. Establecer este directorio es fundamental para mantener una organización clara y un acceso adecuado a los archivos de Tomcat.

---

### INSTALAR UNZIP

Esta tarea instala el paquete unzip en el sistema. El paquete unzip proporciona una utilidad para extraer archivos de archivos comprimidos en formato ZIP. Es necesario para descomprimir archivos descargados, como el archivo de Tomcat, facilitando su instalación y configuración posterior.

---

### INSTALAR TAR

Esta tarea instala el paquete tar en el sistema. El paquete tar proporciona una herramienta para crear y extraer archivos de archivos tar, que son archivos comprimidos comúnmente utilizados para empaquetar varios archivos en uno solo. Instalar tar es esencial para descomprimir archivos de Tomcat si están en formato tar, lo que permite la extracción y configuración del servidor de aplicaciones.

---

### EXTRAER TOMCAT

Esta tarea descomprime el archivo descargado de Tomcat utilizando herramientas como unzip o tar. La extracción del archivo descomprimido coloca los archivos de Tomcat en el directorio de instalación especificado, preparándolos para su configuración y ejecución.

---

### CAMBIAR PERMISOS DEL DIRECTORIO DE TOMCAT

Esta tarea ajusta los permisos del directorio donde se instaló Tomcat. Modificar los permisos asegura que el usuario o grupo adecuado tenga acceso necesario para leer, escribir o ejecutar archivos dentro del directorio, lo que es esencial para la correcta operación del servidor de aplicaciones y para mantener la seguridad del sistema.

---

### CREAR SERVICIO SYSTEMD PARA TOMCAT:

Esta tarea configura un servicio systemd para Tomcat. El servicio systemd permite iniciar, detener y gestionar Tomcat como un servicio en segundo plano en el sistema. Esto asegura que Tomcat se inicie automáticamente al arrancar el sistema y facilite su administración a través de comandos estándar de systemd, mejorando la integración con el sistema operativo.

---

### RECARGAR SYSTEMD

Esta tarea actualiza el sistema para que reconozca cualquier cambio reciente en la configuración de los servicios. Después de crear o modificar un servicio systemd, es necesario recargar la configuración de systemd para aplicar los cambios y permitir que el nuevo o modificado servicio sea gestionado correctamente.

---

### HABILITAR E INICIAR TOMCAT

Esta tarea configura el servicio de Tomcat para que se inicie automáticamente al arrancar el sistema y luego lo inicia de inmediato. Habilitar el servicio asegura que Tomcat esté disponible después de cada reinicio, mientras que iniciar el servicio pone en marcha el servidor de aplicaciones inmediatamente, permitiendo que empiece a funcionar y atender solicitudes.

---

### DESPLEGAR EL ARCHIVO WAR

Esta tarea copia un archivo WAR (Web Application Archive) al directorio de despliegue de Tomcat. El archivo WAR contiene una aplicación web empaquetada, y al desplegarlo, Tomcat lo descomprime y lo configura automáticamente para que la aplicación esté disponible y accesible a través del servidor web.

---

### COPIAR EL ARCHIVO DE CONFIGURACIÓN DE LA APLICACIÓN

Esta tarea transfiere un archivo de configuración específico al directorio adecuado dentro de la instalación de la aplicación. Este archivo puede contener ajustes y parámetros necesarios para la correcta ejecución y configuración de la aplicación en el entorno de Tomcat. Copiar el archivo de configuración es esencial para garantizar que la aplicación funcione de acuerdo con las especificaciones requeridas.

---

### HABILITAR PUERTOS EN EL FIREWALL

Esta tarea configura el firewall del sistema para permitir el tráfico a través de puertos específicos. Esto es necesario para que los servicios de Tomcat o aplicaciones desplegadas puedan recibir solicitudes externas a través de la red. Habilitar los puertos adecuados asegura que los usuarios y otros sistemas puedan comunicarse con el servidor de aplicaciones según sea necesario.

---

### RECARGAR EL FIREWALL

Esta tarea aplica los cambios realizados en la configuración del firewall. Después de habilitar o modificar reglas de firewall, es necesario recargar la configuración para que las nuevas reglas entren en efecto y comiencen a gestionar el tráfico de red de acuerdo con las nuevas directrices.

---

### REINICIAR TOMCAT

Esta tarea reinicia el servicio de tomcat para aplicar los cambios realizados.

```
- name: Reiniciar Tomcat
  service:
    name: tomcat
    state: restarted
```

## PLAYBOOKUBUNTU

---

### CONFIGURAR Y ASEGURAR MARIADB EN UBUNTU

Esta tarea ajusta la configuración y aplica medidas de seguridad para la instalación de MariaDB en un sistema Ubuntu. Esto incluye la configuración inicial del servidor de bases de datos, el establecimiento de contraseñas seguras, la eliminación de usuarios y bases de datos no necesarias, y la configuración de opciones para proteger el acceso y la integridad de los datos. El objetivo es asegurar que MariaDB esté configurado de manera segura y eficiente para su uso en producción.

---

### CONFIGURAR EL INTÉRPRETE DE PYTHON

Esta tarea ajusta la configuración del intérprete de Python en el sistema. Esto puede incluir la instalación de una versión específica de Python, la configuración de variables de entorno, la instalación de paquetes o dependencias necesarios, y la aseguración de que el intérprete esté correctamente configurado para el desarrollo o la ejecución de aplicaciones Python. La configuración adecuada del intérprete garantiza que el entorno de desarrollo o producción funcione sin problemas.

---

### ACTUALIZAR LA LISTA DE PAQUETES

Esta tarea actualiza la base de datos de paquetes del sistema con la información más reciente disponible en los repositorios configurados. Esto asegura que el sistema tenga acceso a la última versión de los paquetes y actualizaciones disponibles, facilitando la instalación de software y actualizaciones seguras y recientes.

---

### INSTALAR MARIADB

Esta tarea instala el servidor de bases de datos MariaDB en el sistema. La instalación de MariaDB proporciona la infraestructura necesaria para crear, gestionar y consultar bases de datos. Esta tarea asegura que el software de MariaDB esté disponible y operativo para su configuración y uso en el entorno de trabajo.

---

### INSTALAR MARIADB Y HERRAMIENTAS RELACIONADAS

Esta tarea instala el servidor de bases de datos MariaDB junto con herramientas adicionales que facilitan la gestión y administración de la base de datos. Las herramientas relacionadas pueden incluir clientes, utilidades para la administración, y otras utilidades que complementan la instalación básica de MariaDB, asegurando que el entorno de bases de datos esté completo y funcional.

---

### VERIFICAR SI MYSQL\_SECURE\_INSTALLATION ESTÁ INSTALADO

Esta tarea comprueba si la herramienta `mysql_secure_installation` está disponible en el sistema. Esta herramienta se utiliza para asegurar la instalación de MariaDB mediante la configuración de opciones de seguridad importantes, como establecer contraseñas de root y eliminar usuarios no necesarios. La verificación garantiza que la herramienta esté instalada antes de intentar usarla.

---

### CAMBIAR BIND

Esta tarea realiza una inserción mediante `ansible.builtin.blockinfile` en el archivo `my.cnf` de MariaDB para agregar el `bind 0.0.0.0` y permitir el acceso remoto desde el servidor de aplicaciones.

```
- name: Cambiar bind
  ansible.builtin.blockinfile:
    path: /etc/mysql/my.cnf
    marker: '# ANSIBLE'
    state: present
    block: |
      [mysqld]
      bind-address = 0.0.0.0
```

---

### AGREGAR DEPENDENCIAS DE PYTHON PARA TAREAS DE MYSQL

Estas dos tareas realizan la instalación de `pip`, `python3-pip` (`pip3`) y `pymysql`. Las dos primeras son dependencias de la última y esta misma es dependencia para la tarea de ejecución del archivo SQL.

```
- name: Instalar pip
  apt:
    name: pip
    state: present

- name: Instalar pip3
  apt:
    name: python3-pip
    state: present

- name: Asegurar pymysql
  become: true
  apt:
    name: python3-pymysql
    state: present
```

---

### MOSTRAR RUTA DE MYSQL\_SECURE\_INSTALLATION

Esta tarea localiza y muestra la ruta completa del comando `mysql_secure_installation` en el sistema. Esto es útil para confirmar que la herramienta está instalada y para saber dónde se encuentra para su uso en procesos de configuración y seguridad de MariaDB.

---

### EJECUTAR MYSQL\_SECURE\_INSTALLATION

Esta tarea ejecuta la herramienta `mysql_secure_installation` para asegurar la instalación de MariaDB. Este proceso configura opciones de seguridad importantes, como establecer la contraseña de root, eliminar usuarios anónimos, deshabilitar el inicio de sesión remoto del usuario root, y eliminar bases de datos de prueba. La ejecución de esta herramienta ayuda a fortalecer la seguridad del servidor de bases de datos.

---

### CREAR DB Y USUARIO PARA LA APP

Esta tarea realiza una copia del archivo `db.sql` al servidor de base de datos y ejecuta dicho archivo en el servicio de MySQL (`mariadb`).

```
- name: Copiar SQL
  ansible.builtin.copy:
    src: "/home/user/obligatorio/ansible/files/db.sql"
    dest: "~/db.sql"

- name: Aplicar SQL
  mysql_db:
    state: import
    login_user: "root"
    login_password: ""
    login_host: "127.0.0.1"
    login_unix_socket: /run/mysqld/mysqld.sock
    name: "todo"
    target: "~/db.sql"
```

---

### REINICIAR SERVICIO MYSQL

Esta tarea reinicia el servicio de MySQL para aplicar los cambios realizados.

```
- name: Reiniciar servicio
  service:
    name: mariadb
    state: restarted
```

## BIBLIOGRAFÍA Y REFERENCIAS

- Documentación de ansible - <https://www.ansible.com>
- Comunidad de stackoverflow (resolución de problemas) - <https://stackoverflow.com/>
- ChatGPT (ejemplos de playbooks) - <https://chatgpt.com/>

