

## Trabajo Práctico – Virtualización con VirtualBox

### **Alumnos:**

Juan Ignacio Resende – juani300199@gmail.com

Ignacio Agustin Tello Salas – Ignacio.tello.dev@gmail.com

**Materia:** Arquitectura y Sistemas Operativos

**Profesor:** Osvaldo Falabella

**Tutora:** Ana Valeria Celerier

**Fecha de Entrega:** 05/06/25

### **1. Introducción**

La virtualización permite ejecutar múltiples sistemas operativos sobre una única máquina física. En este trabajo se exploran conceptos básicos de virtualización y se realiza una instalación y configuración práctica utilizando Oracle VirtualBox. A su vez, se desarrolló un pequeño programa en Python como parte de la práctica final.

### **2. Marco Teórico**

Virtualización de hardware: tecnología que permite correr máquinas virtuales completas sobre un sistema físico (host).

**Hipervisores:**

- Tipo 1 (bare-metal): se ejecutan directamente sobre el hardware.
- Tipo 2 (hosted): se ejecutan sobre un sistema operativo, como VirtualBox.

**Imágenes ISO:** archivos que contienen el contenido de un disco de instalación, como Ubuntu Server.

**Snapshots:** capturas del estado actual de una VM, útiles para retroceder ante errores.

**Redes virtuales:** configuraciones como NAT, Bridge y Host-only permiten simular conexiones reales o aisladas.

### 3. Caso Práctico

Se utilizó Oracle VirtualBox (hipervisor tipo 2) para crear una máquina virtual con Ubuntu Server 22.04 y configurar un servidor web Apache accesible desde la red local.

**Pasos realizados:**

**Crear una nueva máquina virtual:**

**Nombre:** PracticoSeguridad

**Tipo:** Linux

**Versión:** Ubuntu (64-bit)

**Memoria:** 2067 MB

**Disco duro virtual (VDI):** 20 GB (almacenamiento dinámico)

**Montar imagen ISO:**

- Se descargó la ISO oficial desde: <https://ubuntu.com/download/server>
- Se inició la instalación mínima de Ubuntu Server, se creó un usuario y contraseña.

**Instalación del sistema y conexión a red:**

- Durante la instalación se eligió conexión por DHCP (modo Bridge en VirtualBox para acceso externo).

**Instalación del servidor Apache y configuración de firewall:**

- Una vez iniciado el sistema (línea de comandos: usuario@ubuntu:~\$), se ejecutaron:

*sudo apt update*

*sudo apt install apache2*

*sudo ufw allow 'Apache'*

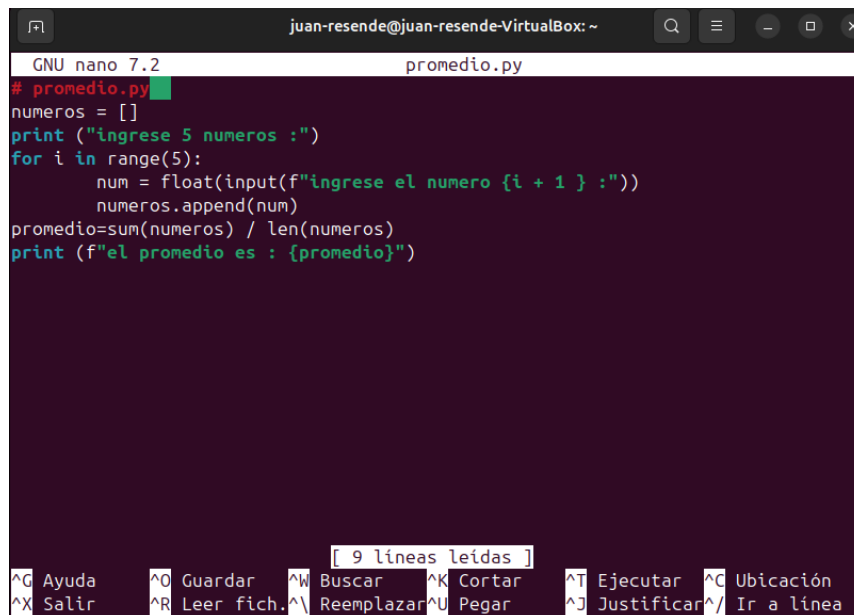
### Creación y ejecución del programa en Python:

- Se verificó que Python estuviera instalado con:  
python3 --version
- Luego se creó el archivo promedio.py con el siguiente contenido:

```

juan-resende@juan-resende-VirtualBox:~$ python3 --version
python 3.12.3
juan-resende@juan-resende-VirtualBox:~$ nano promedio.py

```



```

GNU nano 7.2 promedio.py
# promedio.py
numeros = []
print ("ingrese 5 numeros :")
for i in range(5):
    num = float(input(f"ingrese el numero {i + 1 } :"))
    numeros.append(num)
promedio=sum(numeros) / len(numeros)
print (f"el promedio es : {promedio}")

```

[ 9 líneas leídas ]

^G Ayuda    ^O Guardar    ^W Buscar    ^K Cortar    ^T Ejecutar    ^C Ubicación  
 ^X Salir    ^R Leer fich.    ^\ Reemplazar    ^U Pegar    ^J Justificar    ^\_ Ir a línea

```

juan-resende@juan-resende-VirtualBox:~$ nano promedio.py
juan-resende@juan-resende-VirtualBox:~$ python3 promedio.py
ingrese 5 numeros :
ingrese el numero 1 :9
ingrese el numero 2 :9
ingrese el numero 3 :7
ingrese el numero 4 :8
ingrese el numero 5 :9
el promedio es : 8.4
juan-resende@juan-resende-VirtualBox:~$

```

## 4. Metodología Utilizada

- Instalación de Oracle VirtualBox en Windows.
- Descarga e instalación de Ubuntu Server 22.04 desde ISO.
- Configuración de red virtual en modo Bridge.

### Ejecución de comandos desde consola para configurar servidor Apache.

- Validación del servicio desde otra máquina (o navegador del host) accediendo vía IP de la VM.

## 5. Resultados Obtenidos

- La máquina virtual con Ubuntu Server se ejecutó correctamente en VirtualBox.
- El servidor Apache respondió correctamente desde el navegador usando la IP de la VM.
- Se verificó que el servicio web es accesible por red local y protegido por el firewall.
- Se comprendió el ciclo de instalación, configuración y acceso remoto de un servidor virtualizado.

## 6. Conclusiones

La virtualización simplifica la creación de entornos de prueba y aprendizaje sin necesidad de hardware adicional. VirtualBox es una herramienta accesible y potente para este propósito.

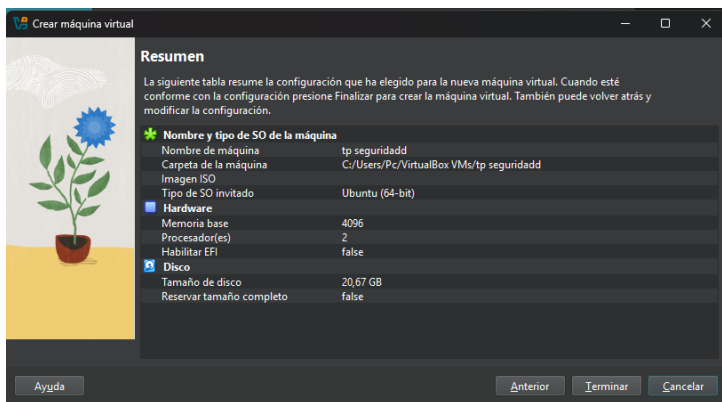
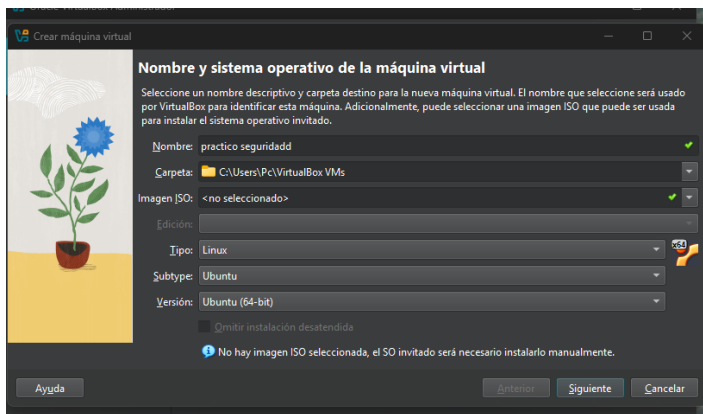
## 7. Bibliografía

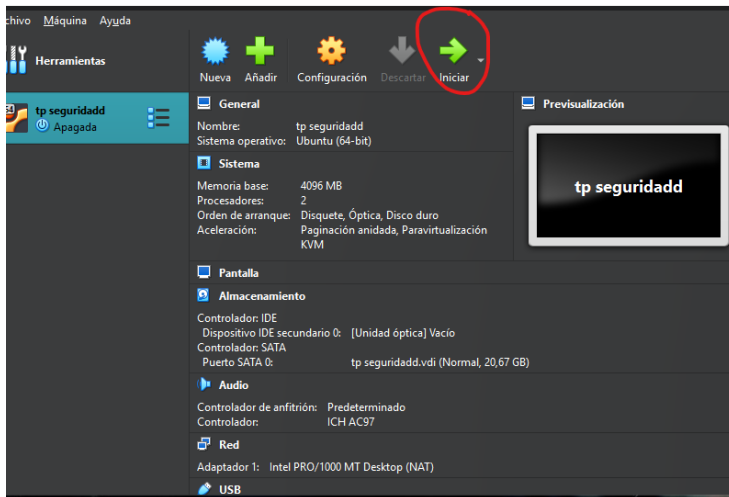
- Video tutorial de la instalacion de ubuntu:  
[https://youtu.be/XfCaHcOdBL8?si=6HTMNq4rFgFgAuL1&utm\\_source=ZTQxO](https://youtu.be/XfCaHcOdBL8?si=6HTMNq4rFgFgAuL1&utm_source=ZTQxO)
- Video explicativo del Profesor Andres Odiard: <https://youtu.be/Ne-IS7m9HEw?si=blyhUQXSg86JbvDk>
- Video explicativo del Profesor Miguel Tola:  
[https://youtu.be/EjySRGAz9CI?si=NnBWEaG5\\_J06ZbZO](https://youtu.be/EjySRGAz9CI?si=NnBWEaG5_J06ZbZO)

- Modelo integrador virtualizacion.
- Get Ubuntu Server: [Get Ubuntu Server | Download | Ubuntu](#)
- Download VirtualBox: <https://www.virtualbox.org/wiki/Downloads>

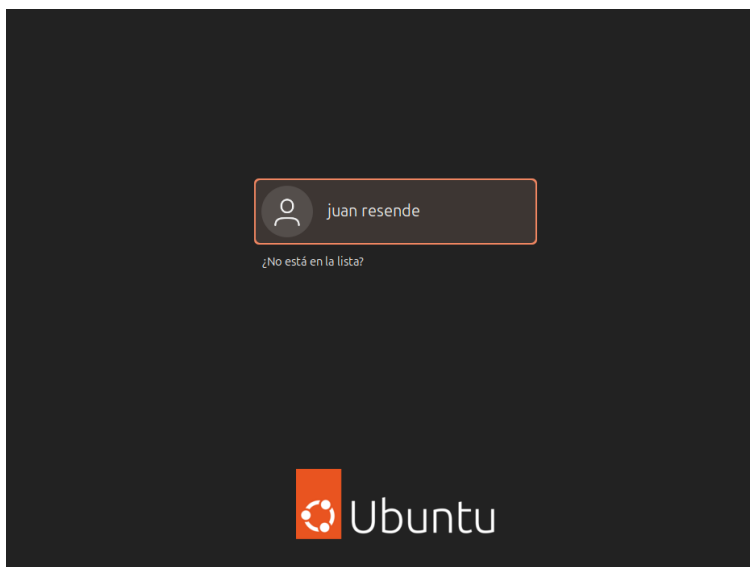
## 8. Anexos:

### Imagenes del proceso de la creacion de la VM:





Una vez iniciado creamos usuario y contraseña:



```
Des:12 http://ar.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages
[1.068 kB]
Des:13 http://ar.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Compon
ts [376 kB]
Des:14 http://ar.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Compon
ents [940 B]
Des:15 http://ar.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components
[7.076 B]
Des:16 http://ar.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Comp
onents [216 B]
Des:17 http://ar.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Compon
ents [16,3 kB]
Des:18 http://ar.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Comp
onents [212 B]
Descargados 3.190 kB en 2s (1.394 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 112 paquetes. Ejecute «apt list --upgradable» para verlos.
juan-resende@juan-resende-VirtualBox:~$ ^C
juan-resende@juan-resende-VirtualBox:~$ sudo apt upgrade -y
```

```

0-1ubuntu7) ...
reparando para desempaquetar .../57-plymouth-theme-spinner_24.004.60-1ubuntu7.1
amd64.deb ...
desempaquetando plymouth-theme-spinner (24.004.60-1ubuntu7.1) sobre (24.004.60-1
ubuntu7) ...
reparando para desempaquetar .../58-plymouth-label_24.004.60-1ubuntu7.1_amd64.d
eb ...
desempaquetando plymouth-label (24.004.60-1ubuntu7.1) sobre (24.004.60-1ubuntu7)
...
reparando para desempaquetar .../59-plymouth_24.004.60-1ubuntu7.1_amd64.deb ...
desempaquetando plymouth (24.004.60-1ubuntu7.1) sobre (24.004.60-1ubuntu7) ...
reparando para desempaquetar .../60-bluez-cups_5.72-0ubuntu5.1_amd64.deb ...
desempaquetando bluez-cups (5.72-0ubuntu5.1) sobre (5.72-0ubuntu5) ...
reparando para desempaquetar .../61-bluez-obexd_5.72-0ubuntu5.1_amd64.deb ...
desempaquetando bluez-obexd (5.72-0ubuntu5.1) sobre (5.72-0ubuntu5) ...
reparando para desempaquetar .../62-dns-root-data_2024071801-ubuntu0.24.04.1_al
.deb ...
desempaquetando dns-root-data (2024071801-ubuntu0.24.04.1) sobre (2023112702-wil
sync1) ...

Progreso: [ 37%] [#####.....]

```

```

juan-resende@juan-resende-VirtualBox: $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:2a:98:c8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.189/24 brd 192.168.100.255 scope global dynamic noprefixrou
te enp0s3
        valid_lft 86355sec preferred_lft 86355sec
    inet6 2803:4fc0:0:c0ea:bd96:ff5b:be5b:afc2/64 scope global temporary dynamic
        valid_lft 259156sec preferred_lft 85866sec
    inet6 2803:4fc0:0:c0ea:a00:27ff:fe2a:98c8/64 scope global dynamic mngtmpaddr
        valid_lft 259156sec preferred_lft 172756sec
    inet6 fe80::a00:27ff:fe2a:98c8/64 scope link
        valid_lft forever preferred_lft forever
juan-resende@juan-resende-VirtualBox: ~$

```

192.168.100.189



## Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dissmod`, `a2ensite`, `a2dissite`, and `a2enconf`. See their respective man pages for detailed information.

