

Actividad 2.

Configuración

Servidor Web

Tejera Santana Adoney

Actividad N.º1 – 2ºA DAW

Curso 2024/2025

Índice

Actividad 2. Configuración Servidor Web.....	1
1.1 Descripción detallada de los trabajos realizados.....	3
1 1. Localizar la carpeta donde se publican las páginas web en el Servidor Apache en /var/www/html/. Crear una página con el siguiente contenido:.....	3
2 Crear una carpeta en /var/www/html/ que se llame “actividades” y crear dos archivos con el siguiente código.....	4
3 Acceder a la carpeta de los archivos de configuración de Apache e indicar las carpetas y archivos que aparecen. Aportar captura de pantalla y diferenciarlo por los colores que establece.....	5
4 Configurar el número máximo de peticiones o solicitudes que puede atender el servidor Apache y el tiempo máximo que pueden estar activas. Para hacer esta parte vamos a subir la memoria RAM de la máquina virtual a 4GB.....	6
5 Monitorizar el uso de los recursos del servidor web con las herramientas que proporciona Apache.....	7
6 Mostrar las páginas web creadas desde el navegador de la máquina anfitrión.....	8
1. Instalar Docker engine en Ubuntu.....	10
2. Instalar contenedor con Apache, MySQL y PHP y realizar pruebas de funcionamiento	13
3. Instalar contenedor con Tomcat y realizar pruebas de funcionamiento.....	14
1.2 Herramientas empleadas.....	15
1.3 Problemática encontrada y solución.....	15
1.4 Conclusiones.....	15



1.1 Descripción detallada de los trabajos realizados

1. CONFIGURACIÓN APACHE

1. Localizar la carpeta donde se publican las páginas web en el Servidor Apache en /var/www/html/. Crear una página con el siguiente contenido:

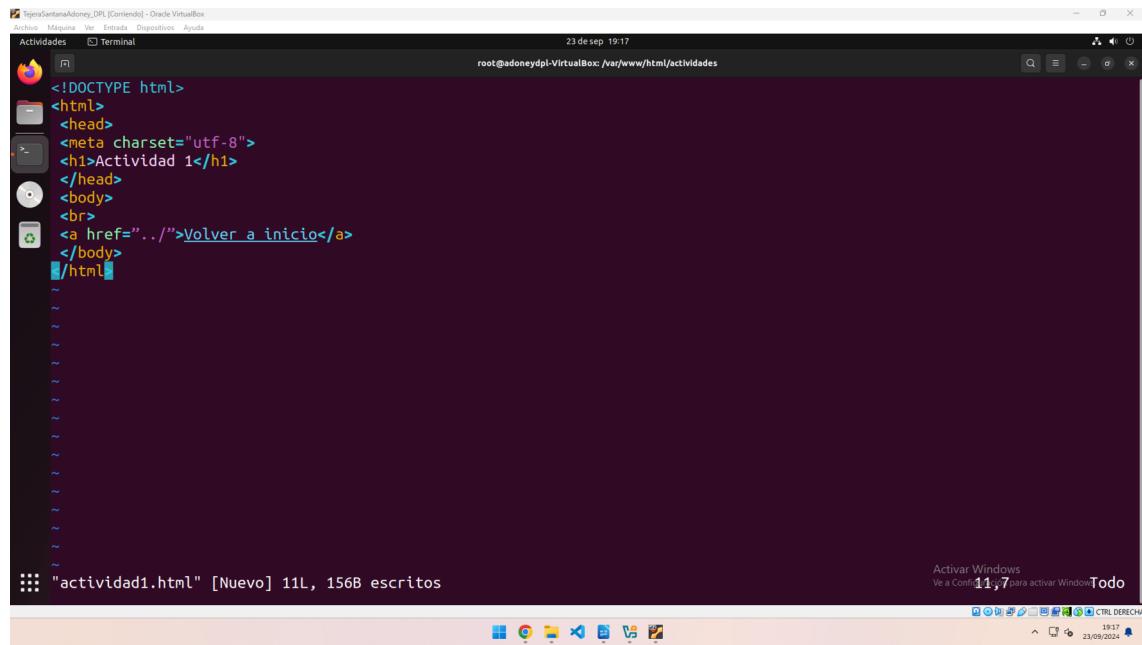
The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "TejeraSantanaAdoney_DPL [Conectado] - Oracle VirtualBox". The terminal command is "root@adoneydpl-VirtualBox: /var/www/html>". The code displayed in the terminal is:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Despliegue de Aplicaciones Web - DPL</title>
</head>
<body>
<br>
<a href=".//actividades/actividad1.html">Actividad 1</a>
<br>
<a href=".//actividades/actividad2.html">Actividad 2</a>
</body>
</html>
```

The desktop environment includes a dock with icons for various applications like a browser, file manager, and terminal. The taskbar at the bottom shows the date and time (23/sep 19:13), system status (14, 0 ~ 1 activar Window Todo), and a battery icon.

Se crea la página

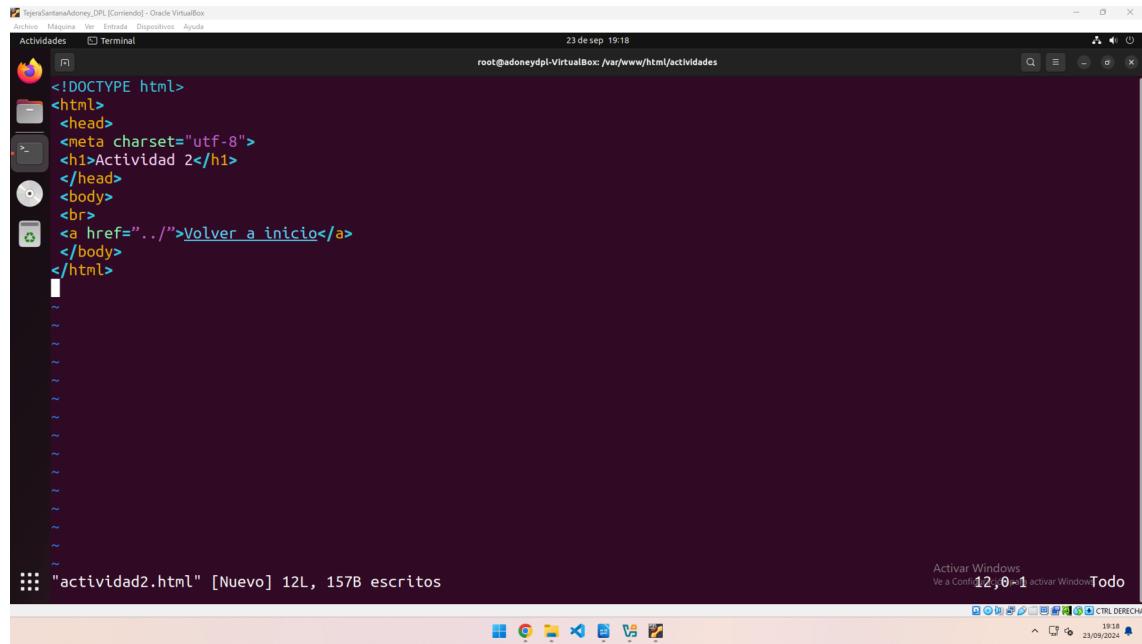
2 Crear una carpeta en /var/www/html/ que se llame "actividades" y crear dos archivos con el siguiente código



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<h1>Actividad 1</h1>
</head>
<body>
<br>
<a href=".">Volver a inicio</a>
</body>
</html>
```

"actividad1.html" [Nuevo] 11L, 156B escritos

Creamos el archivo "Actividad1.html"

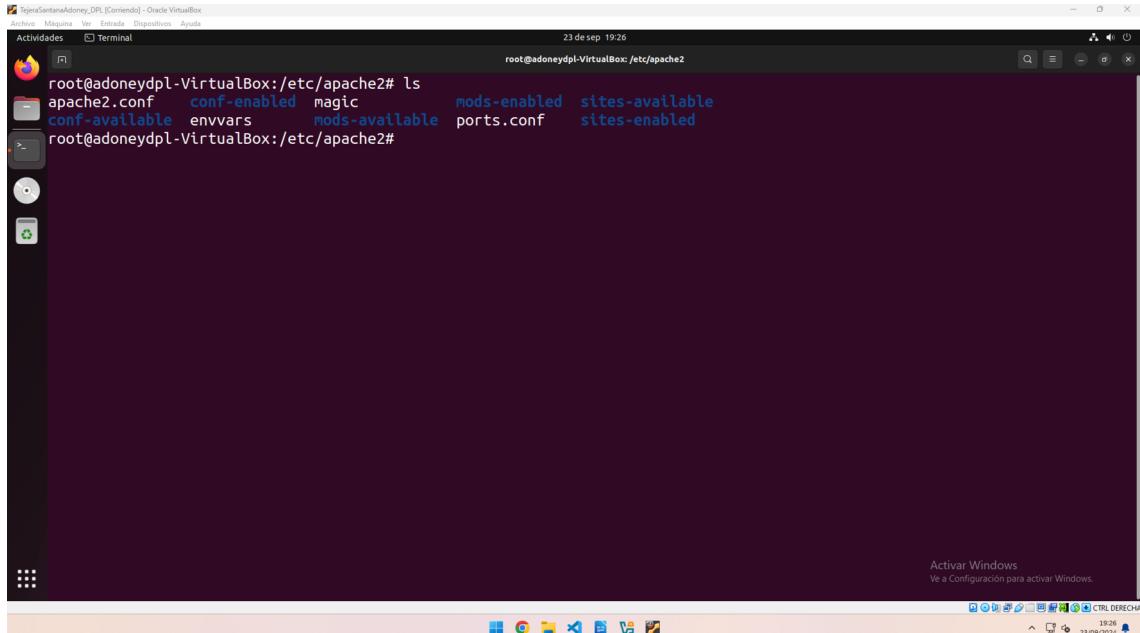


```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<h1>Actividad 2</h1>
</head>
<body>
<br>
<a href=".">Volver a inicio</a>
</body>
</html>
```

"actividad2.html" [Nuevo] 12L, 157B escritos

Y la actividad2.html

3 Acceder a la carpeta de los archivos de configuración de Apache e indicar las carpetas y archivos que aparecen. Aportar captura de pantalla y diferenciarlo por los colores que establece.

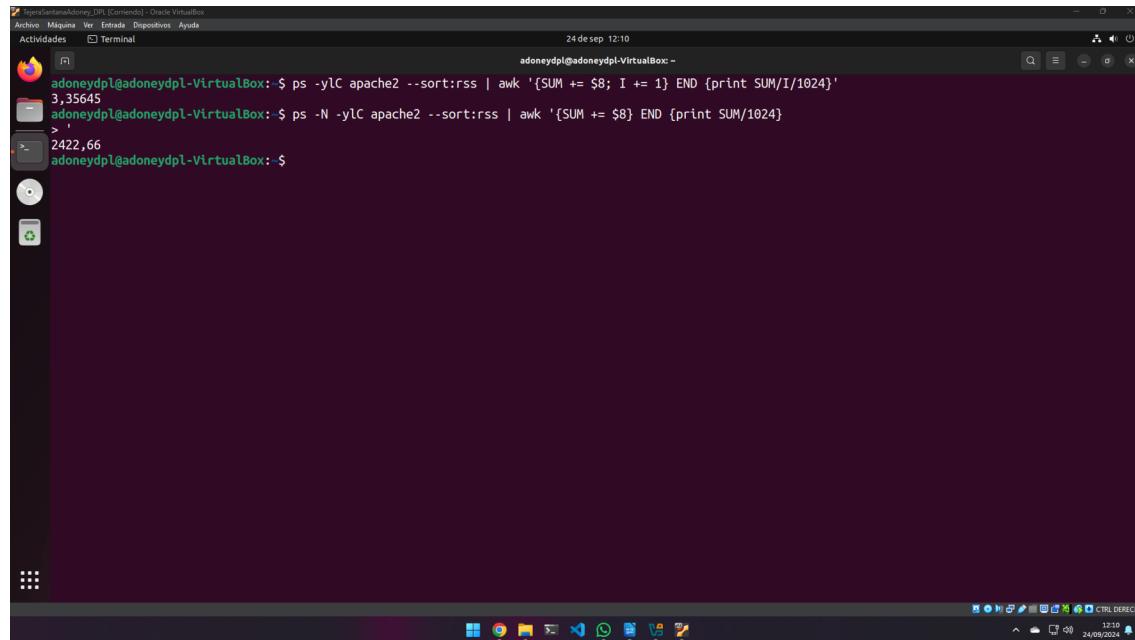


```
root@adoneydpl-VirtualBox:/etc/apache2# ls
apache2.conf  conf-enabled  magic      mods-available  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
root@adoneydpl-VirtualBox:/etc/apache2#
```

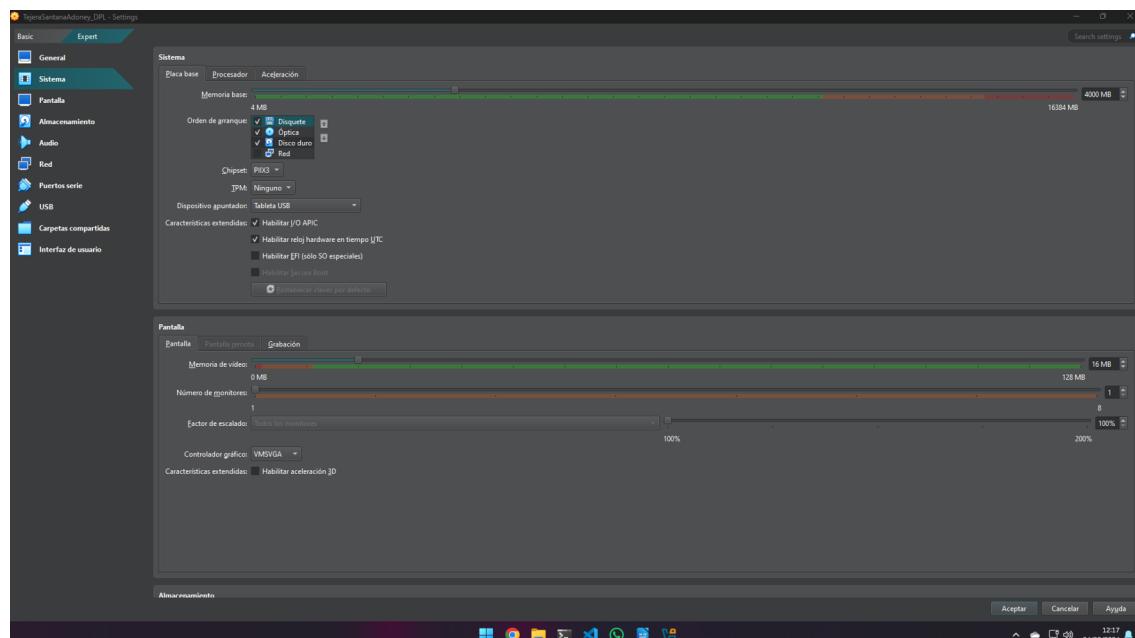
Accedemos a la carpeta y se lista los archivos

- apache2.conf: configuración principal de apache.
- Magic: ayuda al servidor a reconocer qué tipo de archivo está manejando
- ports.conf: configuración de los puertos del servidor
- envvars: contiene las variables de entorno de apache
- /conf-enable: contiene configuración adicional de apache activado
- /conf-available: configuración adicional de apache (no todos estan activos)
- /mods-available: contiene todos los módulos instalados en el servidor que puedes usar
- /mods-enable: contiene un enlace a los módulos que están activos
- /sites-available: configuración de sitios disponibles
- /sites-enable: contiene enlace a los archivos de configuración sitios disponibles (en sites-available)

4 Configurar el número máximo de peticiones o solicitudes que puede atender el servidor Apache y el tiempo máximo que pueden estar activas. Para hacer esta parte vamos a subir la memoria RAM de la máquina virtual a 4GB.



Se comprueba la cantidad de ram utilizada por apache y posteriormente del resto de procesos. $n^o\text{Conexiones} = \frac{\text{RAMtotal} - \text{RAMresto}}{\text{RAM por conexión}} = \frac{11200 - 2422,66}{3,35} = 2.620,1$



Se disminuye a 4GB de RAM a la máquina virtual.

```

adoneydpl@adoneydpl-VirtualBox:~$ ps -ylc apache2 --sort:rss | awk '{SUM += $8; I += 1} END {print SUM/I}1024'
3,38867
adoneydpl@adoneydpl-VirtualBox:~$ ps -N -ylc apache2 --sort:rss | awk '{SUM += $8} END {print SUM/1024}'
2597,8
adoneydpl@adoneydpl-VirtualBox:~$ 

```

Volvemos a comprobar el consumo y se vuelve a calcular el máximo número de conexiones.

$$n^{\circ}\text{Conexiones} = \frac{\text{RAMtotal} - \text{RAMresto}}{\text{RAM por conexión}} = \frac{4000 - 2597,8}{3,38} = 414,85 \text{ conexiones}$$

5 Monitorizar el uso de los recursos del servidor web con las herramientas que proporciona Apache

Slot	PID	Stopping	Connections	Threads	Async connections				
			total	accepting	busy	idle	writing	keep-alive	closing
0	899	no	0	yes	1	24	0	0	0
1	900	no	0	yes	0	25	0	0	0
Sum	2	0	0	0	1	49	0	0	0
	*								

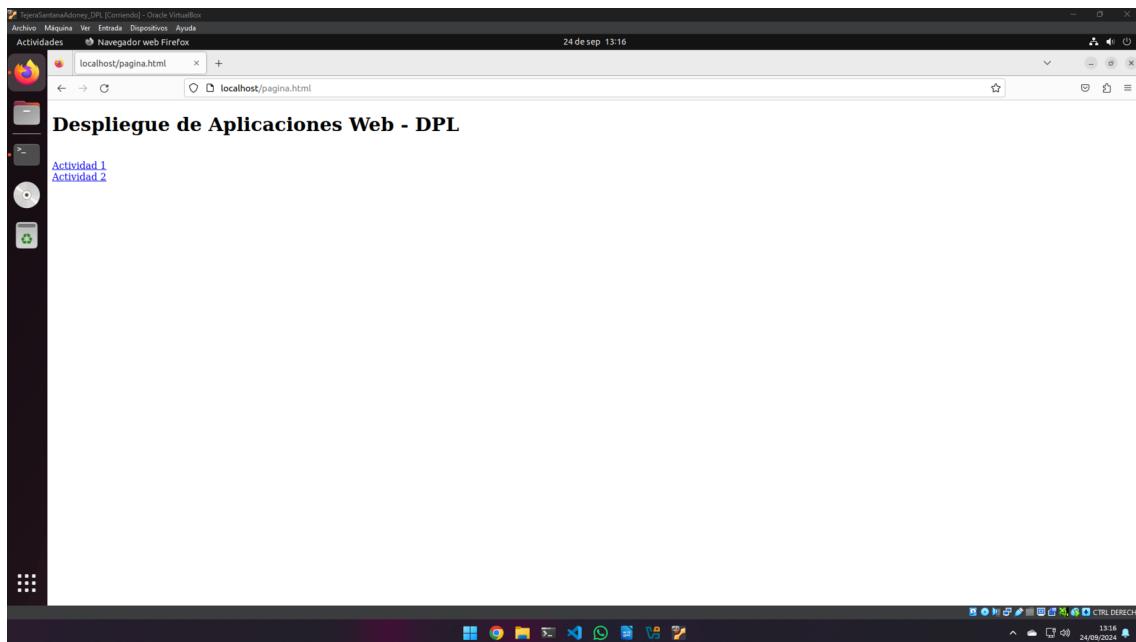
Scoreboard Key:
 * Waiting for Connection, * Starting up, * Reading Request,
 * Writing Request, * Keeping Alive, * DNS Lookup,
 * Closing Connection, * Logging, * Gracefully finishing,
 * File cleanup of worker, * Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
0-0	899	0/1/1	0.02	12	18	0	0.0	0.00	0.00	127.0.0.1	http://1.1	127.0.1.1:80	GET / HTTP/1.1	
0-0	899	0/1/1	0.02	10	3	3	0.0	0.00	0.00	127.0.0.1	http://1.1	127.0.1.1:80	GET /icons/ubuntu-logo.png	HTTP/1.1
0-0	899	0/1/1	0.02	10	0	0	0.0	0.00	0.00	127.0.0.1	http://1.1	127.0.1.1:80	GET /favicon.ico	HTTP/1.1
0-0	899	1/0/0	0.00	0.00	0	0	0.00	0.00	0.00	127.0.0.1	http://1.1	127.0.1.1:80	GET /server-status	HTTP/1.1

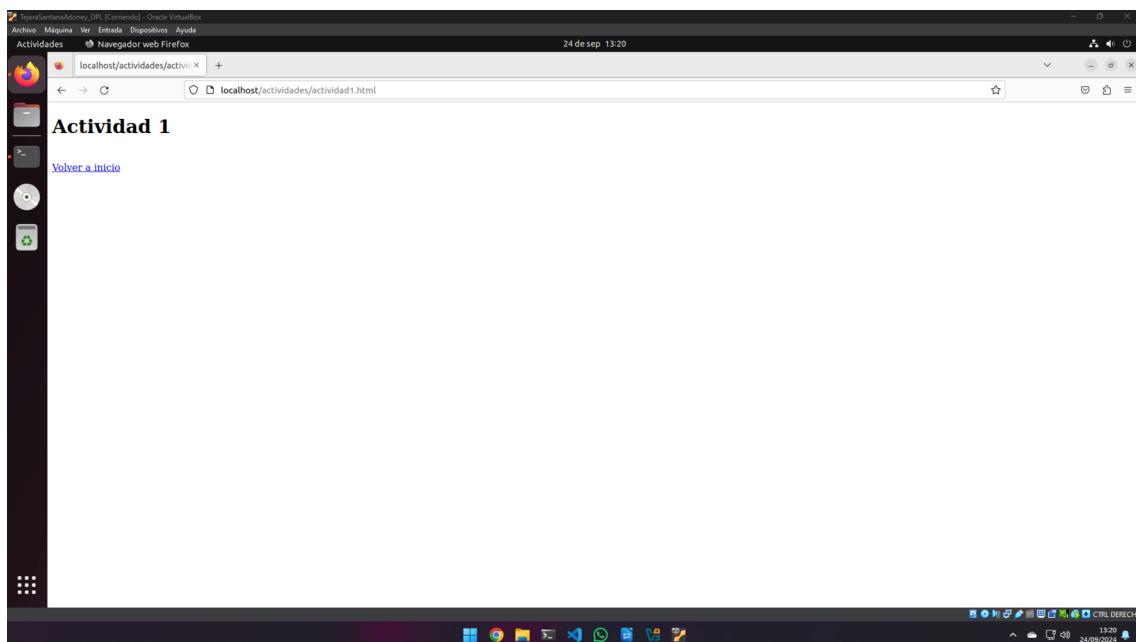
Srv Child Server number - generation
 PID OS process ID

Se monitorea el servidor, abajo se pueden ver los procesos en ejecución.

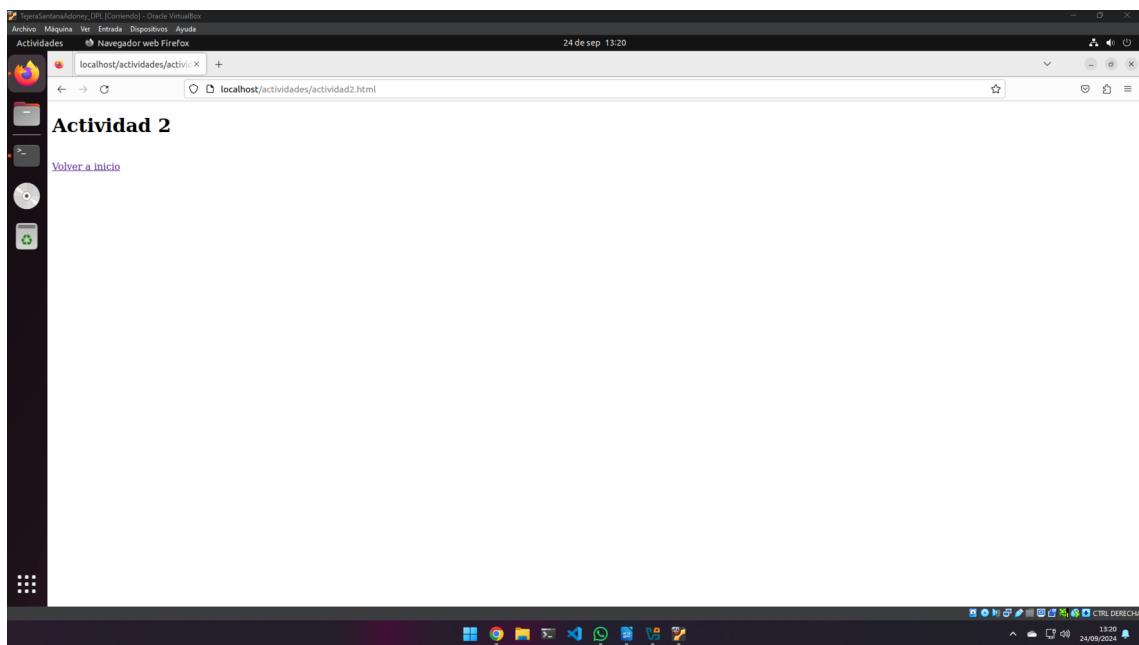
6 Mostrar las páginas web creadas desde el navegador de la máquina anfitrión.



Vista de la página principal.



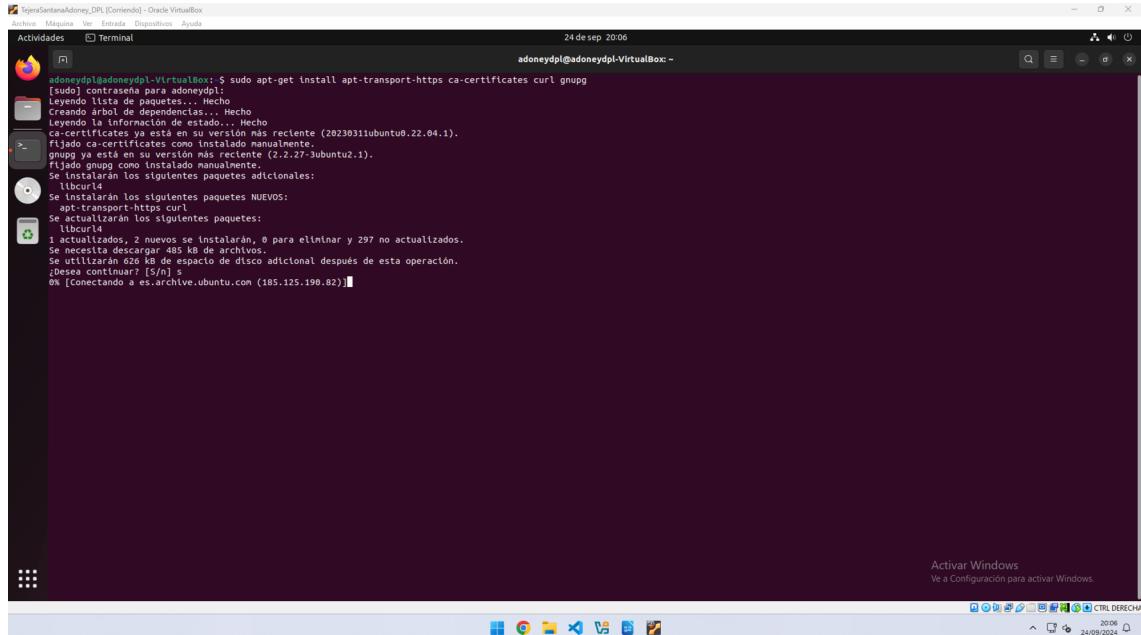
Página de actividad 1



Y finalmente página de actividad 2

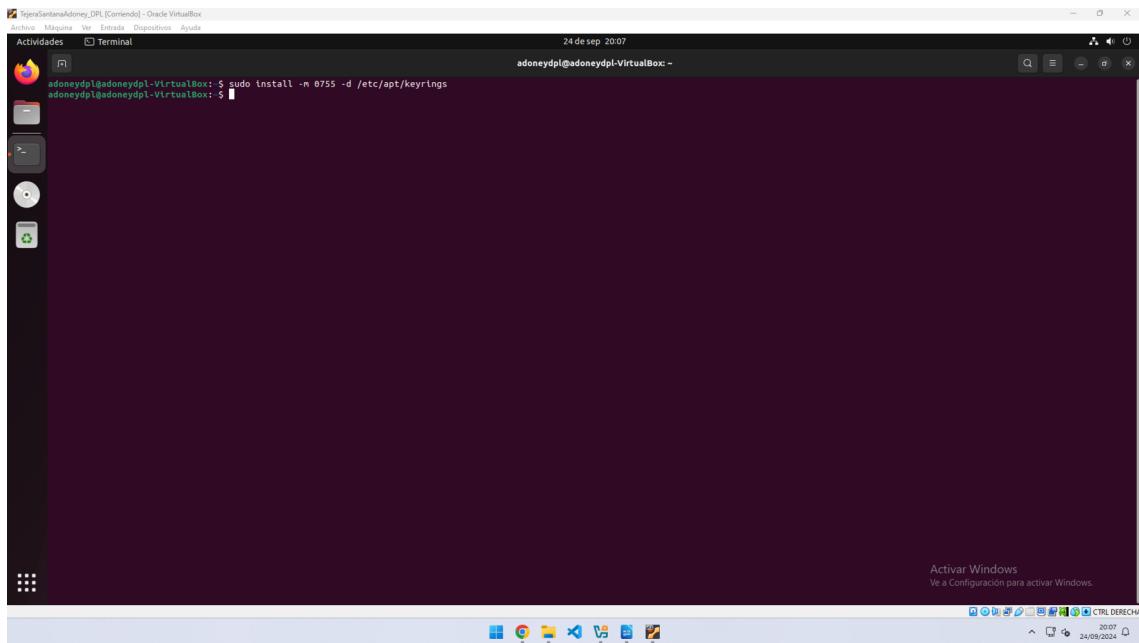
1.2 Despliegue Web con tecnología de virtualización en contenedores

1. Instalar Docker engine en Ubuntu.

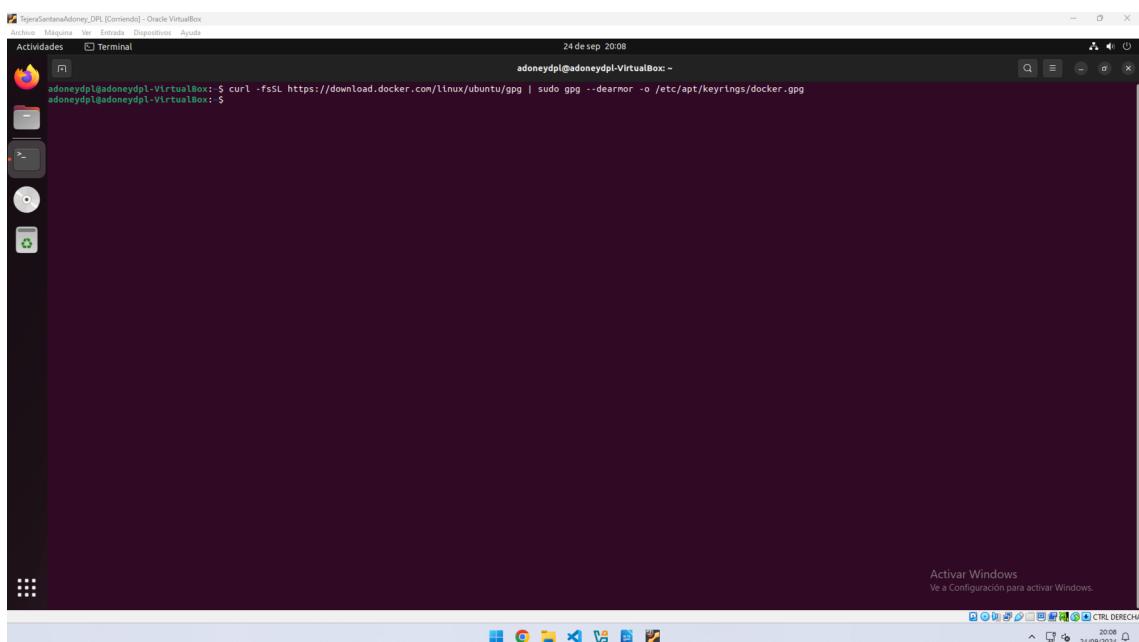


```
adoneydpl@adoneydpl-VirtualBox: ~
[sudo] contraseña para adoneydpl:
Leyendo lista de paquetes... Hecho
Cargando información de estado... Hecho
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20230311ubuntu0.22.04.1).
gnupg ya está en su versión más reciente (2.2.27-3ubuntu2.1).
gnupg ya está en su versión más reciente (2.2.27-3ubuntu2.1).
gnupg como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
  libcurl4
  Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https curl
  Se actualizarán los siguientes paquetes:
  libcurl4
  1 actualizado, 2 nuevos se instalarán, 0 para eliminar y 297 no actualizados.
  Se necesita descargar 485 kB de archivos.
  Se utilizarán 626 kB de espacio de disco adicional después de esta operación.
  ¿Desea continuar? [S/n] s
  0% [Conectando a es.archive.ubuntu.com (185.125.190.82)]
```

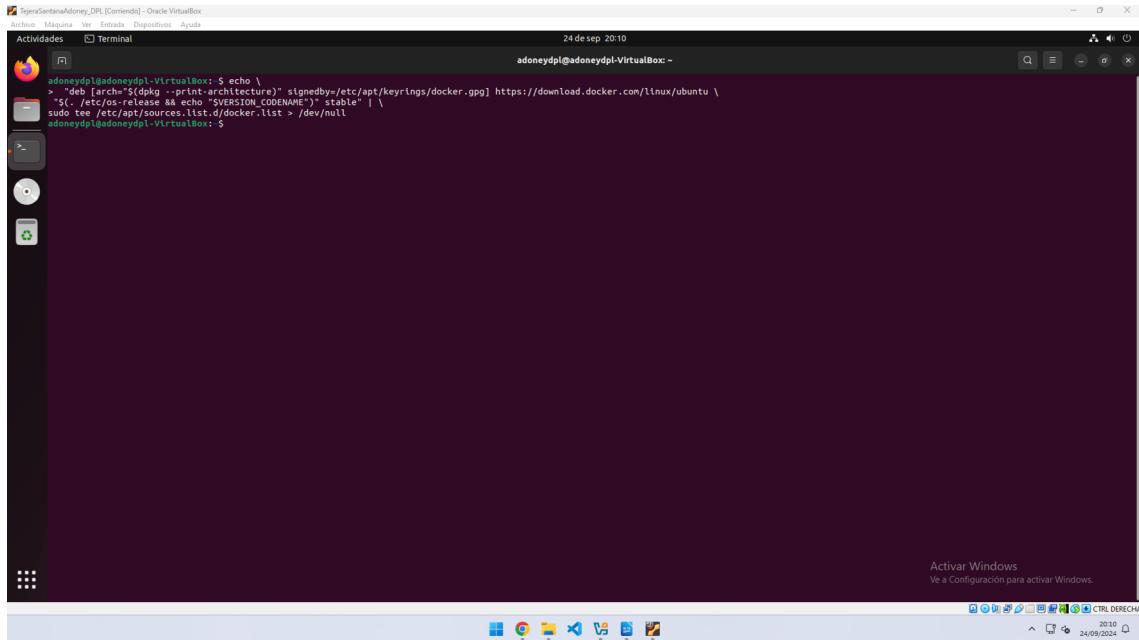
Primero se deberá instalar los paquetes de intercambio de información por HTTP.



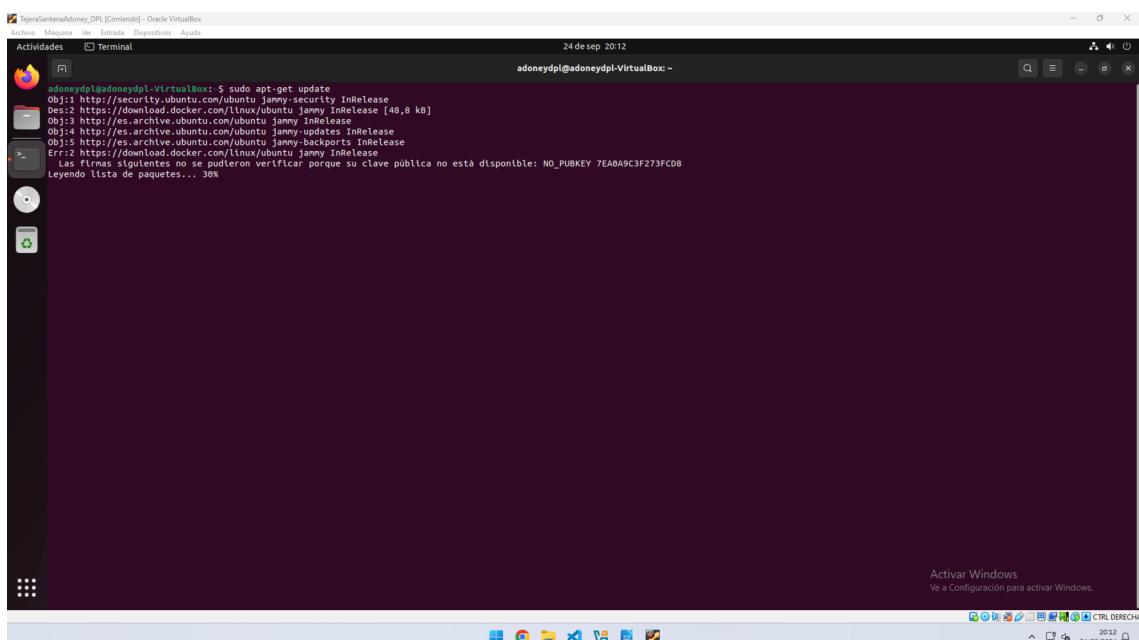
Añadimos el directorio para la clave para poder descargar docker.



Descargamos la clave



Añadimos el repositorio al administrador de paquetes APT.



Actualizamos la lista de paquetes.

```
adoneydpl@adoneydpl-VirtualBox:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Levantando información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  docker-ce-rootless-extras git git-man liblberror-perl libslirp0 pigz slrp4netns
Paquetes sugeridos:
  liblberrror-perl libslirp0 pigz slrp4netns
Se instalarán los siguientes paquetes NUEVOS:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras docker-compose-plugin git git-man liblberrror-perl libslirp0 pigz slrp4netns
  0 actualizados, 12 nuevos se instalarán, 0 para eliminar, 0 ya están actualizados.
  Se necesita descargar 127 MB de archivos.
  Se extraerán 127 MB de los archivos.
  ¿Desea continuar? [S/n] s
Des1: http://es.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63,6 kB]
Des2: https://download.docker.com/linux/ubuntu/jammy/stable/amd64 containerd.io_1.7.22-1_amd64.deb [29,5 MB]
Des3: https://download.docker.com/linux/ubuntu/jammy/stable/amd64 docker_20.10.12~3-0~ubu2204_all.deb [26,5 kB]
Des4: http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1~ubuntui.11 [555 kB]
4% [4 git-man 2.533 8/955 kB 0%] [2 containerd.io 1.048 kB/29,5 MB 4%]
```

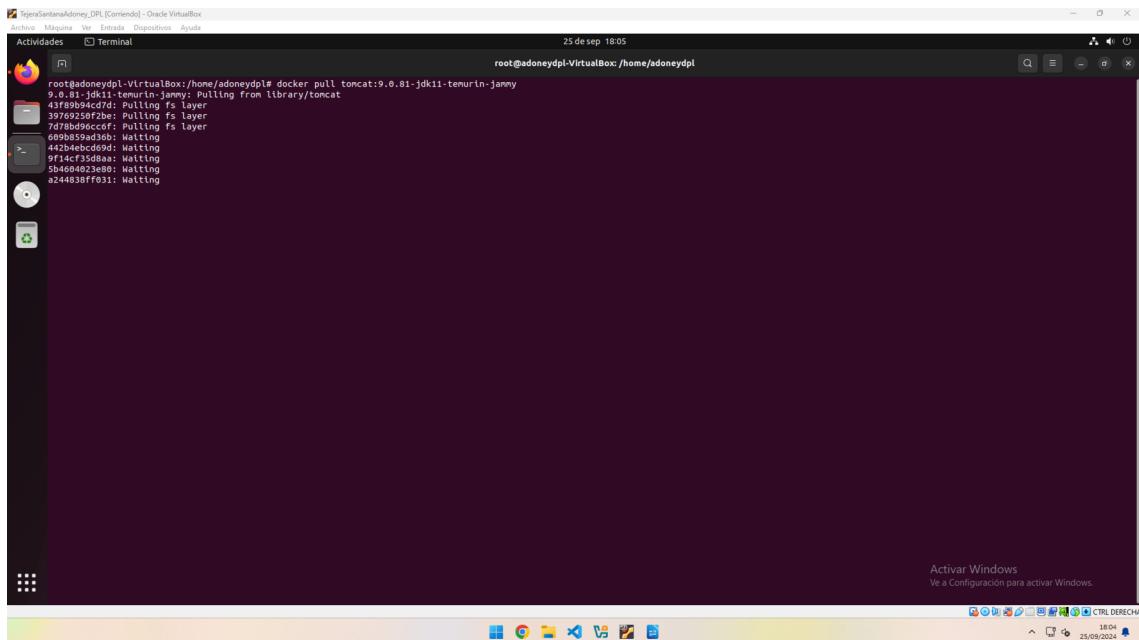
Ya podemos empezar a instalar docker

2. Instalar contenedor con Apache, MySQL y PHP y realizar pruebas de funcionamiento

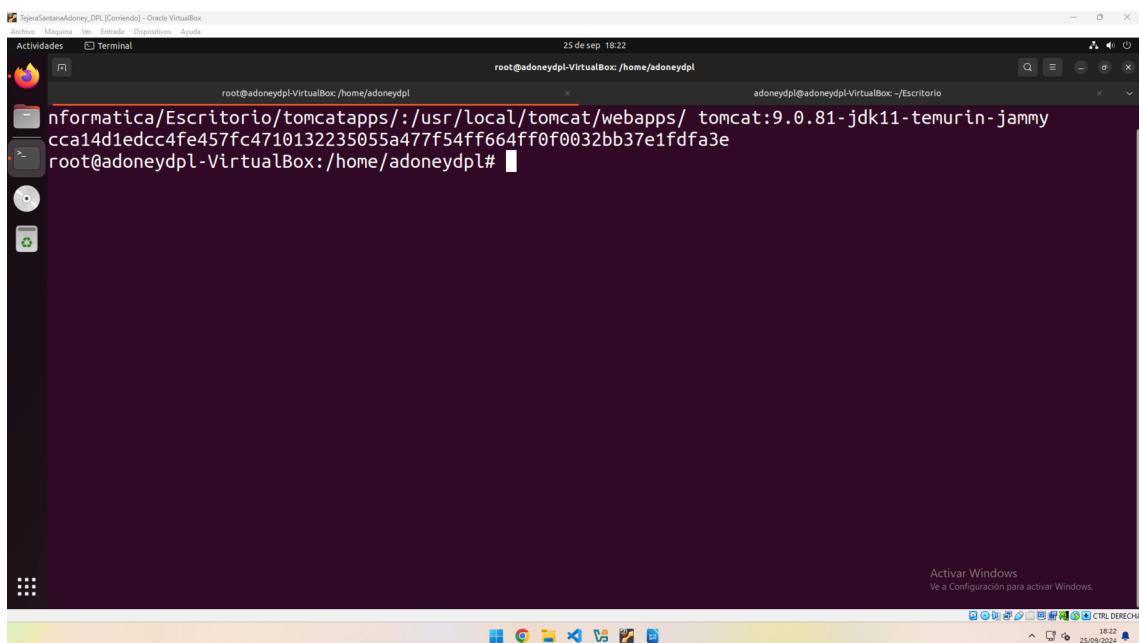
```
adoneydpl@adoneydpl-VirtualBox:~$ sudo docker pull pensiero/apache-php-mysql
Using default tag: latest
```

Descargamos la imagen de docket hub

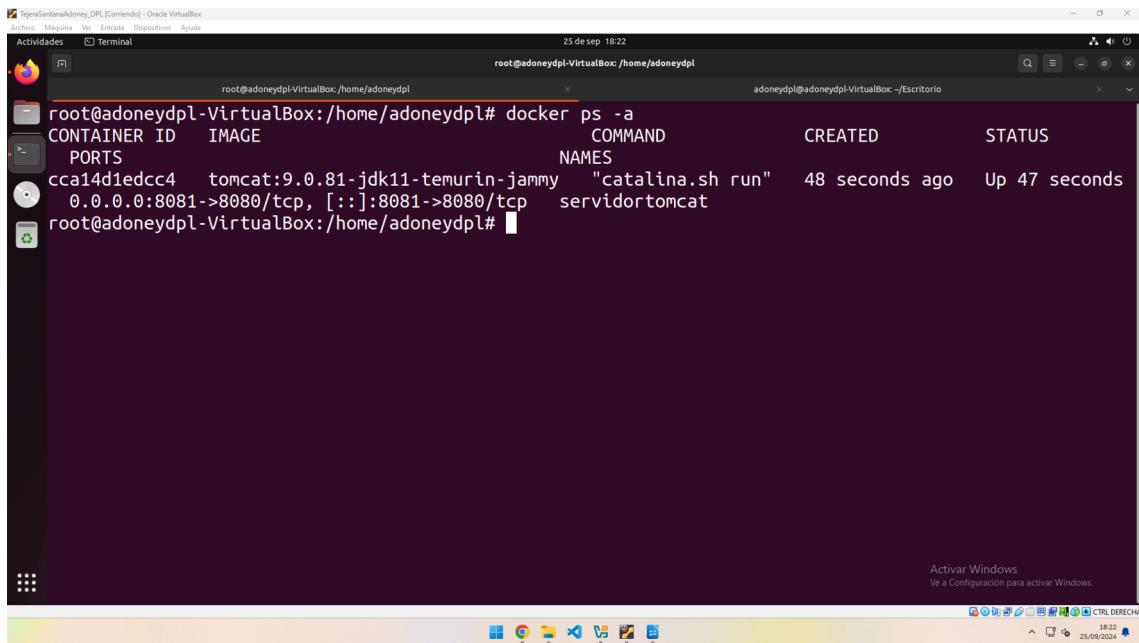
3. Instalar contenedor con Tomcat y realizar pruebas de funcionamiento.



Descargando Tomcat.



Se crea el contenedor.



Comprobamos que se haya creado.

1.2 Herramientas empleadas

En esta actividad hemos hecho uso del ubuntu 22.04, junto a las herramientas web de apache, docker y java Tomcat

1.3 Problemática encontrada y solución

1.4 Conclusiones

En resumen, hemos aprendido a integrar Apache, Docker y Java Tomcat, los cuales permiten crear un entorno de desarrollo y despliegue ágil, escalable y eficiente, facilitando la gestión de servicios y aplicaciones web de manera modular y portátil.

