



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

INGENIERÍA EN SOFTWARE

DESARROLLO WEB AVANZADO

Tarea #: 2

ESTUDIANTE: Jairo Smith Bonilla Hidalgo


UNIDAD: #1

NRC: 23275

Sangolquí, 26 del 2025

CONTENIDO

INTRODUCCION	3
OBJETIVOS	3
DESARROLLO	4
1. Instalación de wamp(uniserver) en Windows	4
2. Instalación de rocky Linux.....	6
3. Instalación de un servidor apache	9
4. Publicar un hola mundo en el servidor apache	10
5. Configuración de de archivo php.ini	13
6. Instalar Docker.....	14
7. Ejercicio de dokerizar una web	17
8. Publicar un proyecto en docker hub.....	18
9. Instalar el docker de un compañero de sus aulas.....	20
10. Instalar la base de datos postgres	21
CONCLUSIONES.....	22
PARA MI.....	22
BIBLIOGRAFIA	23

	Instalación aplicaciones en servidor web		Página 3 de 23
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

INTRODUCCION

En el contexto del desarrollo y despliegue de aplicaciones web y servicios en entornos controlados, es fundamental comprender y manejar diversas herramientas y entornos de servidor. Este informe presenta una serie de actividades orientadas a la instalación, configuración y uso de tecnologías clave para la administración de servidores y contenedores.

Se inicia con la instalación de entornos de servidor como WAMP (o UniServer) en Windows y Rocky Linux como sistema operativo alternativo orientado a servidores. A partir de allí, se procede con la instalación y configuración de Apache HTTP Server, uno de los servidores web más utilizados a nivel mundial, así como la personalización del archivo `php.ini` para gestionar la ejecución de scripts PHP.

Posteriormente, se abordan tareas modernas en el ámbito del desarrollo web mediante el uso de Docker, una tecnología de contenedores que permite la creación de entornos reproducibles, portables y eficientes. El informe incluye la creación y "dockerización" de una aplicación web, su publicación en Docker Hub y la instalación de una imagen creada por otro compañero. Finalmente, se documenta el proceso de instalación de la base de datos PostgreSQL, ampliamente utilizada por su robustez y rendimiento.

Todo el conjunto de actividades busca no solo aplicar conocimientos técnicos, sino también reforzar competencias clave en el manejo de infraestructuras de desarrollo y producción en el área de la informática.


OBJETIVOS

Objetivo General:

- Adquirir y aplicar conocimientos prácticos sobre la instalación, configuración y uso de servidores, contenedores Docker y bases de datos, mediante entornos de desarrollo tanto en Windows como en Linux.

Objetivos Específicos:

- Instalar correctamente un servidor WAMP o UniServer en Windows y configurar su entorno de ejecución.
- Instalar el sistema operativo Rocky Linux y familiarizarse con su entorno de administración.
- Implementar un servidor web Apache y desplegar una aplicación básica de prueba tipo "Hola Mundo".
- Configurar parámetros esenciales del archivo `php.ini` para ajustar el comportamiento del intérprete de PHP.
- Instalar Docker y comprender su funcionamiento en la creación y gestión de contenedores.
- Crear una imagen Docker para una aplicación web personalizada y publicarla en Docker Hub.
- Descargar e implementar un contenedor Docker desarrollado por otro estudiante.

	Instalación aplicaciones en servidor web	Página 4 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- Instalar y verificar el correcto funcionamiento del sistema de gestión de bases de datos PostgreSQL.

DESARROLLO

1. Instalación de wamp(uniserver) en Windows

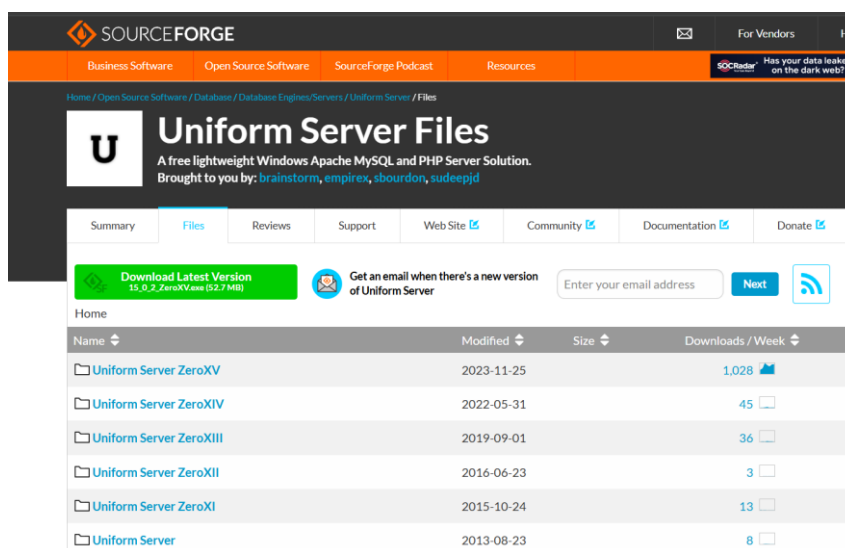
- **Descargar UniServer**


Para instalar UniServer, primero se debe acceder al sitio oficial del proyecto:

<https://www.uniformserver.com>



En esta página, se selecciona la opción **"Download"** y se escoge la versión más reciente de UniServer o la versión recomendada disponible. El archivo descargado será un comprimido .zip que contiene el servidor listo para ejecutar sin instalación adicional.

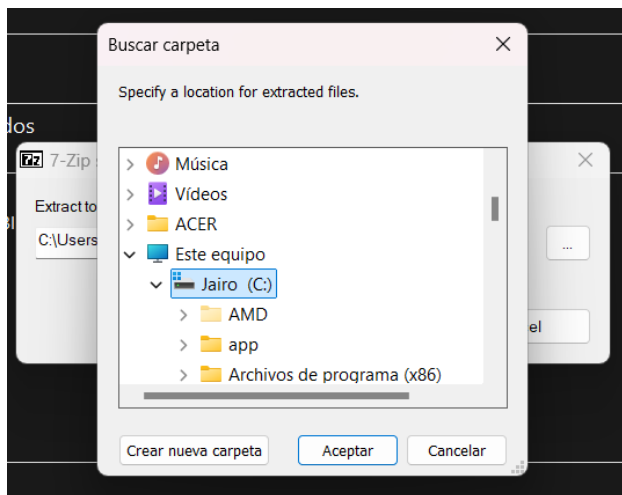


	Instalación aplicaciones en servidor web		Página 5 de 23
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

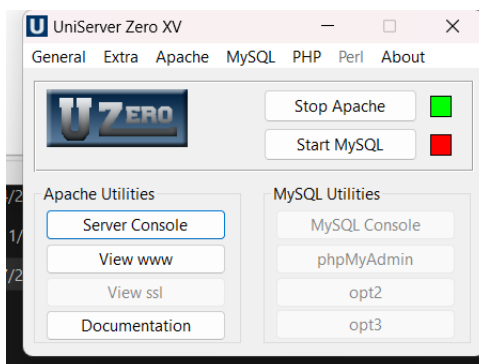
- **Extraer y ejecutar UniServer**

Una vez descargado el archivo, se debe extraer todo su contenido en una ubicación preferida del disco, como, por ejemplo:

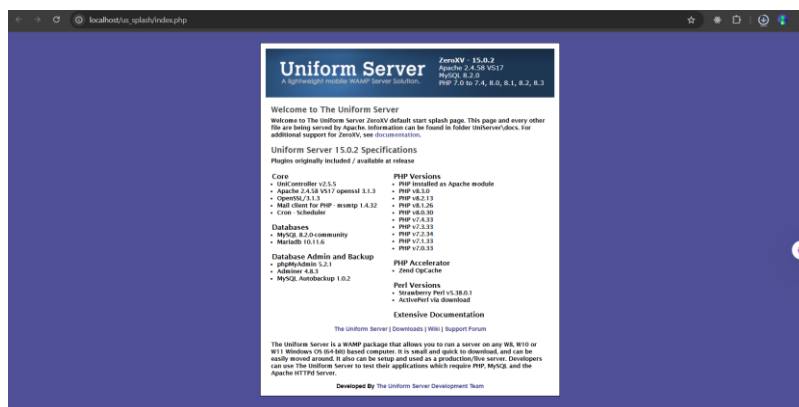
C:\UniServerZ




- Dentro de la carpeta extraída, se debe abrir el UniController y luego iniciar el servicio de Apache.

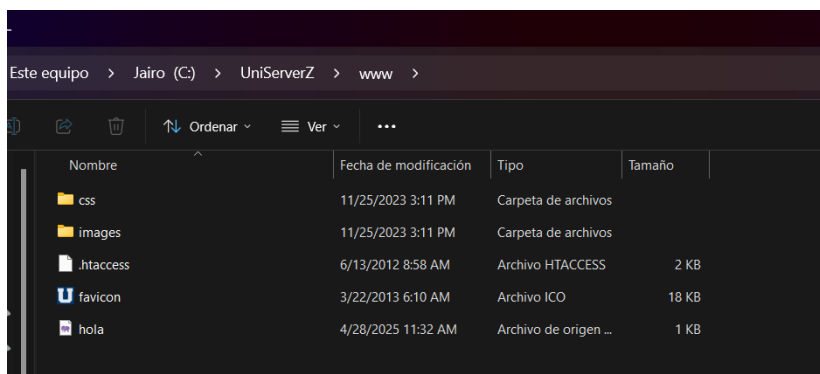


- Se abrirá automáticamente una pestaña en el navegador con la interfaz de control del servidor.



	Instalación aplicaciones en servidor web		Página 6 de 23
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- Se crea un archivo PHP dentro de la carpeta **www** para alojar el código del proyecto o página web.

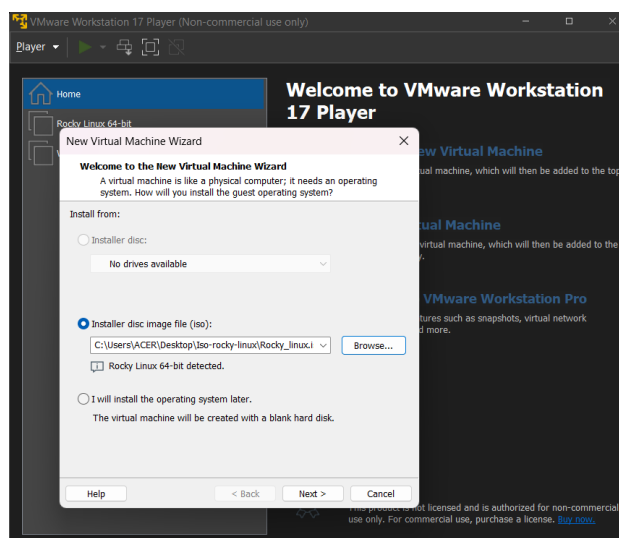



- Abrir la siguiente ruta <http://localhost/hola.php> para comprobar que se haya instalado correctamente:



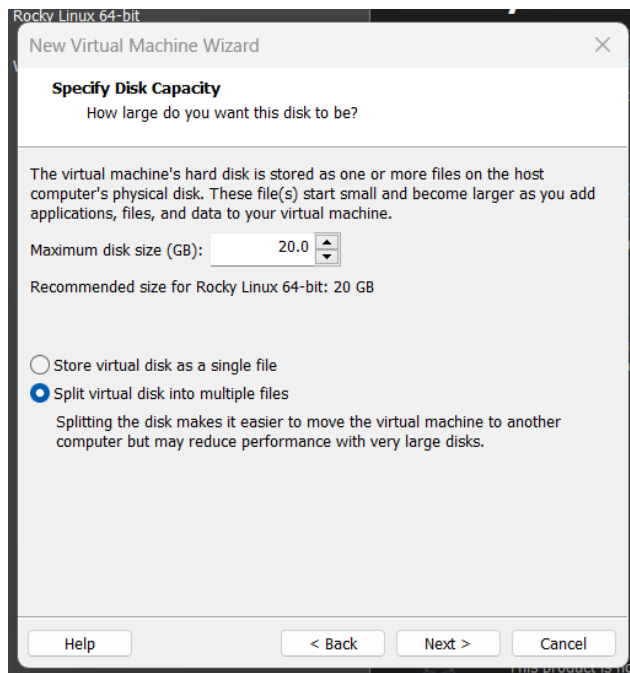
2. Instalación de rocky Linux

- Se debe contar previamente con una máquina virtual y la imagen ISO de Rocky Linux. Luego, se procede a crear una nueva máquina virtual y se selecciona dicha imagen como medio de instalación.

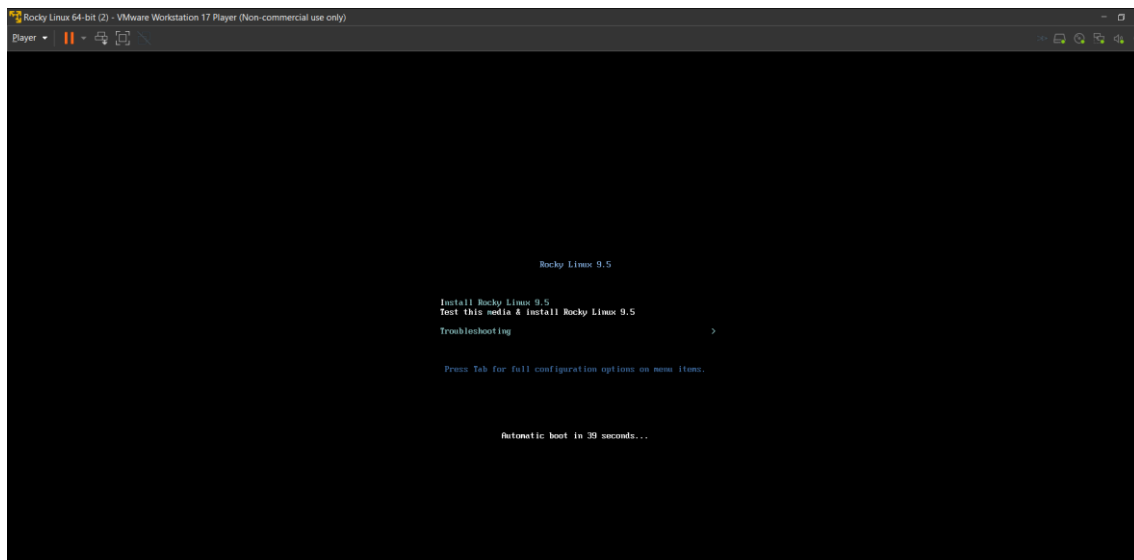



	Instalación aplicaciones en servidor web	Página 7 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- Se asigna un tamaño máximo de disco, en este caso 20 GB, y se continúa con la configuración haciendo clic en "Next".



- Al mostrarse la siguiente ventana del instalador, se procede a continuar presionando la tecla Enter.



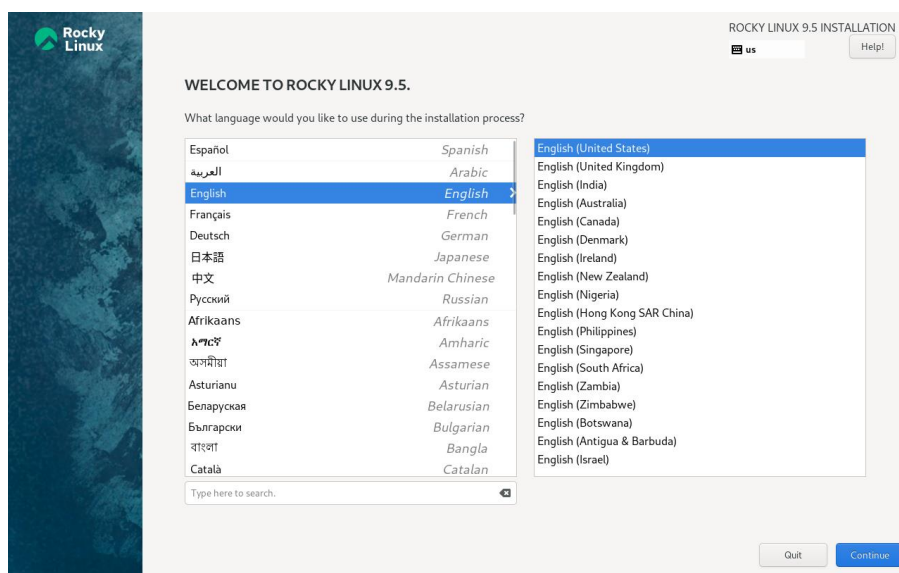
	Instalación aplicaciones en servidor web	Página 8 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- Se espera a que finalice el proceso de instalación de los componentes del sistema.

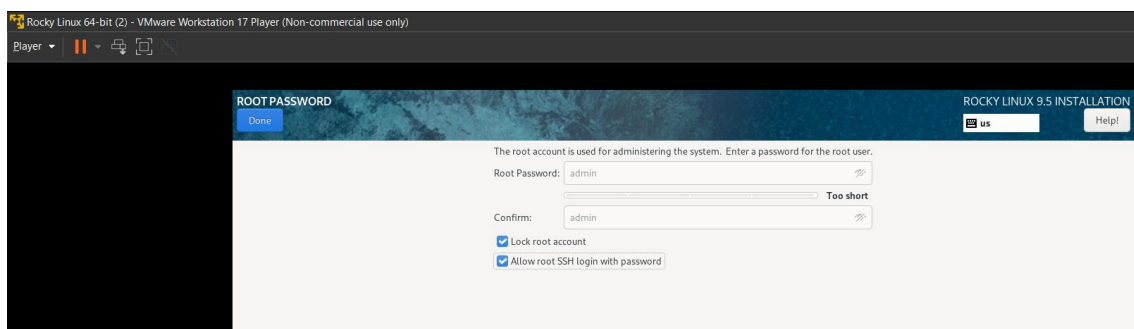
```
Rocky Linux 64-bit (2) - VMware Workstation 17 Player (Non-commercial use only)


[ OK ] Started Show Plymouth Boot Screen.
[ OK ] Started Forward Password Requests to Plymouth Directory Watch.
[ OK ] Reached target Local Encrypted Volumes.
[ OK ] Reached target Path Units.
[ OK ] Started cancel waiting for multipath siblings of nvme0n1.
[ OK ] Finished Wait for udev To Complete Device Initialization.
Starting Device-Mapper Multipath Device Controller...
[ OK ] Started Device-Mapper Multipath Device Controller.
[ OK ] Reached target Preparation for Local File Systems.
[ OK ] Reached target Local File Systems.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Basic System.
/dev/sr0: 4a987c36c841eb2a233cb924b8815ddd
Fragment sums: 9e7b24ab4d541fc357972c7f17c6ffbc6d58efd2ccc915dd164cedf24ac2
Fragment count: 28
Supported ISO: yes
Press [Esc] to abort check.
Checking: 889.5%
```

- Se selecciona el idioma del sistema, eligiendo en este caso inglés como idioma principal.

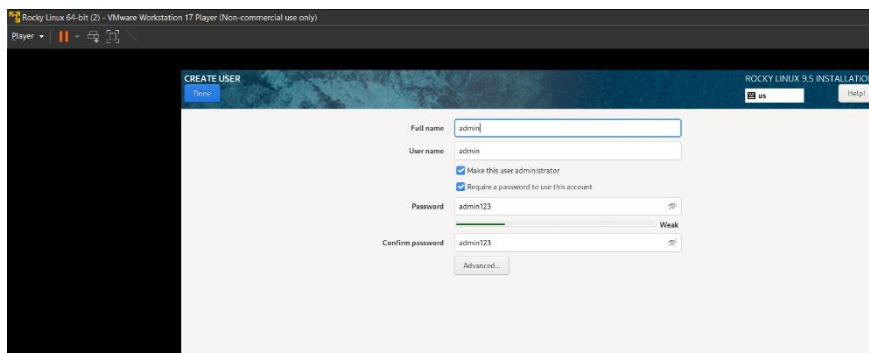


- Se establece una contraseña

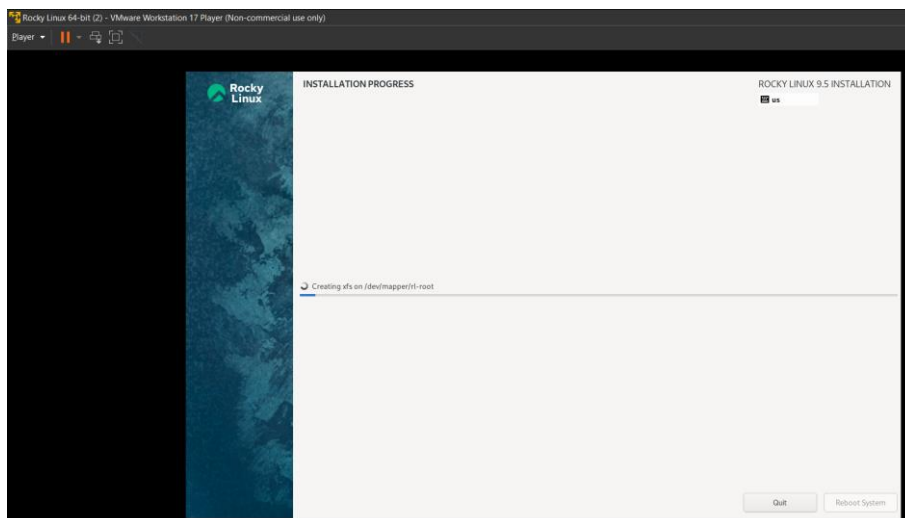


	Instalación aplicaciones en servidor web	Página 9 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- Se crea el usuario



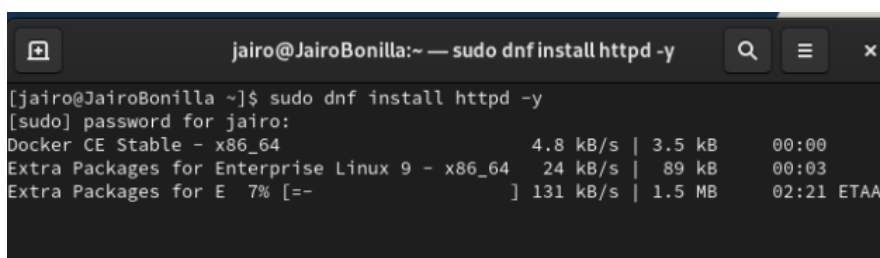
- Una vez que se ha configurado todo lo necesario, se hace clic en **Iniciar instalación** y se espera a que el proceso finalice.




3. Instalación de un servidor apache

- **Instalar Apache (httpd):**

`sudo dnf install httpd -y`



	Instalación aplicaciones en servidor web	Página 10 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- **Habilitar Apache para que inicie automáticamente al arrancar el sistema**

sudo systemctl enable httpd

```
jairo@JairoBonilla ~]$ sudo systemctl enable httpd
[sudo] password for jairo:
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
jairo@JairoBonilla ~]$
```

- **Verificar que Apache está corriendo correctamente**

sudo systemctl status httpd

```
jairo@JairoBonilla:~ — sudo systemctl status httpd
jairo@JairoBonilla ~]$ sudo systemctl status httpd
[sudo] password for jairo:
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Sat 2025-05-24 20:41:40 -05; 2h 24min ago
     Docs: man:httpd.service(8)
  Main PID: 4383 (httpd)
   Status: "Total requests: 8; Idle/Busy workers 100/0; Requests/sec: 0.000922"
    Tasks: 230 (limit: 10751)
   Memory: 8.6M
      CPU: 11.866s
   CGroup: /system.slice/httpd.service
            └─4383 /usr/sbin/httpd -DFOREGROUND
              └─4386 /usr/sbin/httpd -DFOREGROUND
                └─4387 /usr/sbin/httpd -DFOREGROUND
                  └─4388 /usr/sbin/httpd -DFOREGROUND
                    └─4389 /usr/sbin/httpd -DFOREGROUND
                      └─8076 /usr/sbin/httpd -DFOREGROUND

ay 24 20:41:15 JairoBonilla systemd[1]: Starting The Apache HTTP Server...
ay 24 20:41:30 JairoBonilla httpd[4383]: AH00558: httpd: Could not reliably de
ay 24 20:41:40 JairoBonilla systemd[1]: Started The Apache HTTP Server.
ay 24 20:41:40 JairoBonilla httpd[4383]: Server configured, listening on: port
```

- **Iniciar el servicio de Apache en el sistema operativo.**

sudo systemctl start httpd


```
[jairo@JairoBonilla ~]$ sudo systemctl start httpd
[jairo@JairoBonilla ~]$
```

4. Publicar un hola mundo en el servidor apache

- **Acceder y crear un archivo nuevo en la carpeta raíz del servidor web**

sudo nano /var/www/html/hola.html

```
[root@JairoBonilla jairo]# nano /var/www/html/hola.html
```

	Instalación aplicaciones en servidor web	Página 11 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

- Escribir el código para mostrar "Hola Mundo" y guardar



```

GNU nano 5.6.1 /var/www/html/hola.html
<!DOCTYPE html>
<html>
<head>
  <title>Hola Mundo</title>
</head>
<body>
  <h1>¡Hola Mundo desde Apache!</h1>
</body>
</html>

```


- Abrir el navegador y acceder a la página



¡Hola Mundo desde Apache!

- Ahora ejecutar un hola mundo con php


```
cd /var/www/html
sudo nano hola.php
```



```

jairo@JairoBonilla:~$ cd /var/www/html
[jairo@JairoBonilla html]$ sudo nano hola.php
[sudo] password for jairo:
[jairo@JairoBonilla html]$

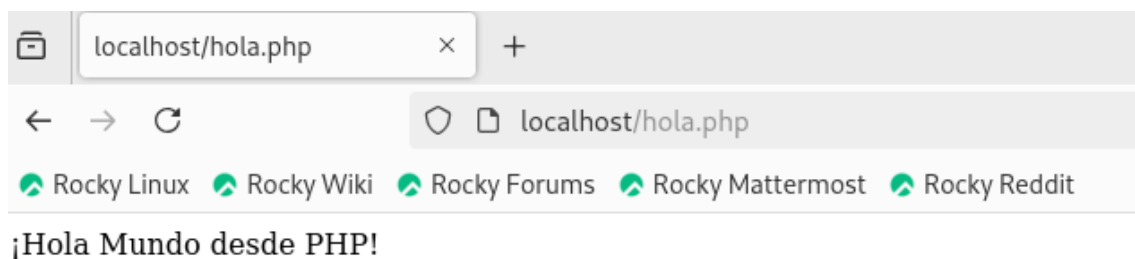
```

	Instalación aplicaciones en servidor web	Página 12 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

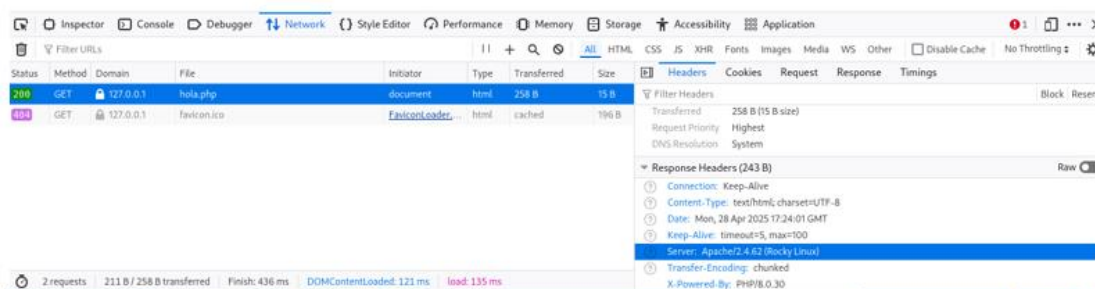
Copiar el siguiente script en el archivo hola.php




- Guardar y cerrar el archivo (Ctrl + O, Enter para guardar en nano, luego Ctrl + X para salir).
- Acceder al siguiente link: <http://localhost/hola.php> y se debe ver lo siguiente:



Darle seguridad a la página



	Instalación aplicaciones en servidor web	Página 13 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

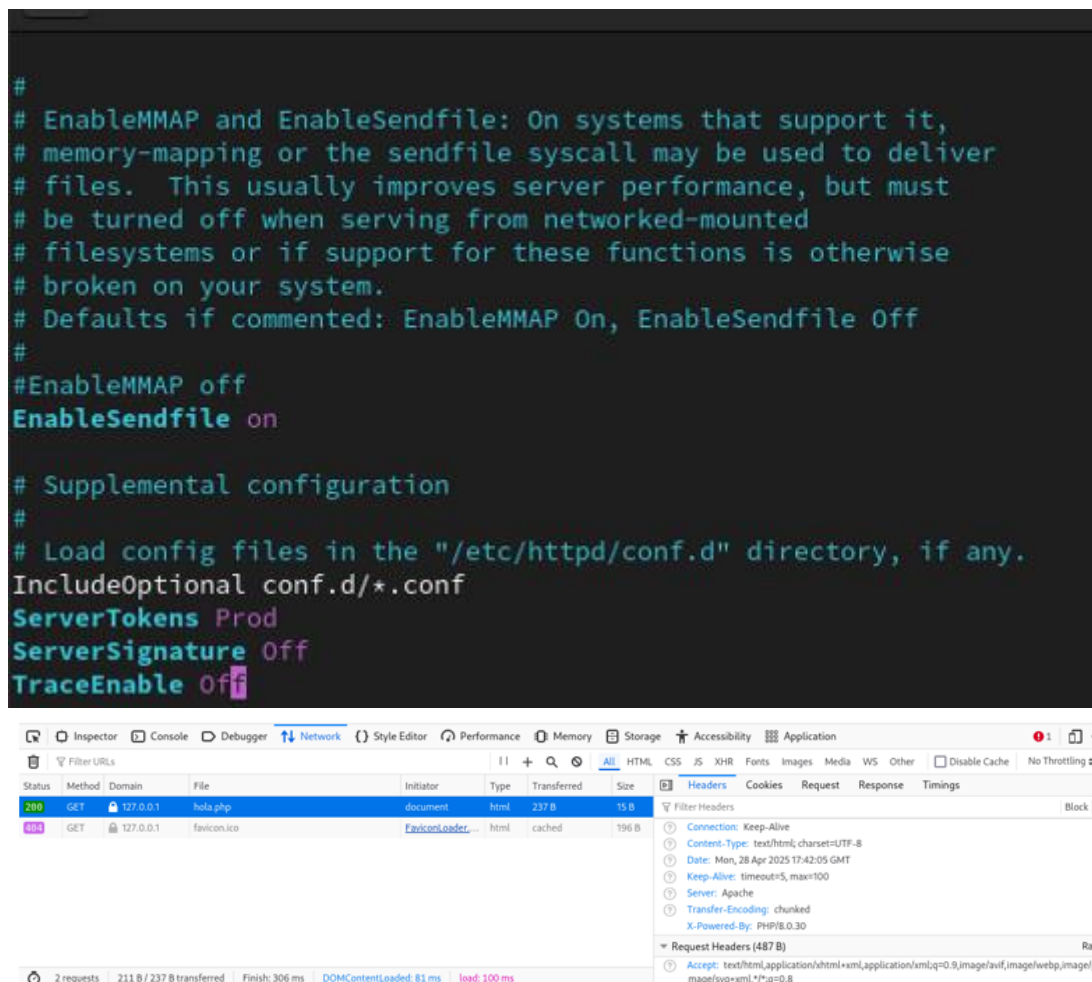
- Ejecutar los siguientes comandos:

vi /etc/httpd/conf/httpd.conf

ServerTokens Prod no muestre config servidor

ServerSignature Off

TraceEnable Off



```

#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
ServerTokens Prod
ServerSignature Off
TraceEnable Off

```

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	127.0.0.1	hola.php	document	html	237 B	15 B
200	GET	127.0.0.1	favicon.ico	Favicon loader...	html	cached	196 B


2 requests | 211 B / 237 B transferred | Finish: 306 ms | DOMContentLoaded: 81 ms | load: 100 ms

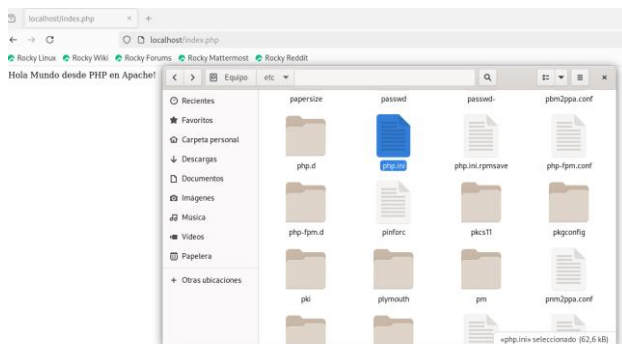
Request Headers (487 B)

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,jpeg/svg+xml,*/*;q=0.8

5. Configuración de de archivo php.ini

- Acceder a la siguiente ruta /etc/php.ini

	Instalación aplicaciones en servidor web		Página 14 de 23
	Desarrollo web avanzado		Unidad: 1
	Bonilla Hidalgo Jairo Smith		NRC: 23275



- Cambiar “short_open_tag = On” en php.ini y reiniciar Apache para usar etiquetas cortas <? ?>.

```
197 ; http://php.net/short-open-tag
198 short_open_tag = ON
```

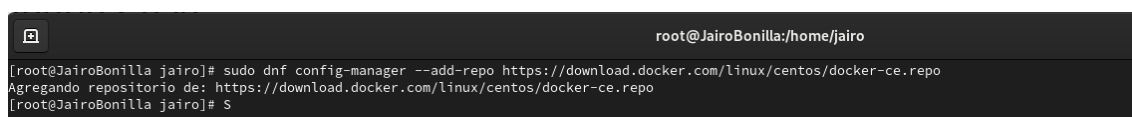
- Se cambia “display_errors = Off” en php.ini y se reinicia Apache para ocultar los errores en producción.

```
501 ; Production Value: Offs
502 ; http://php.net/display-errors
503 display_errors = Off
```

6. Instalar Docker


- Añadir el repositorio de Docker para tienda de aplicaciones

dnf config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>



- Instalar Docker ce, cli, contenedor, build y compose

dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin dockercompose-plugin

	Instalación aplicaciones en servidor web		Página 15 de 23	
	Desarrollo web avanzado		Unidad:	1
	Bonilla Hidalgo Jairo Smith		NRC:	23275

```

root@JairoBonilla:/home/jairo
[root@JairoBonilla jairo]# sudo dnf install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Docker CE Stable - x86_64
kB/s | 70 kB  00:01
Dependencias resueltas.
=====
Paquete                                Arquitectura  Versión      Repositorio
-----
Instalando:
  containerd.io                        x86_64       1.7.27-3.1.el9  docker-ce-stable
  docker-buildx-plugin                 x86_64       0.23.0-1.el9   docker-ce-stable
  docker-ce                            x86_64       3:28.1.1-1.el9  docker-ce-stable
  docker-ce-cli                       x86_64       1:28.1.1-1.el9  docker-ce-stable
  docker-compose-plugin                x86_64       2.35.1-1.el9   docker-ce-stable
Instalando dependencias débiles:
  docker-ce-rootless-extras           x86_64       28.1.1-1.el9   docker-ce-stable
3.2 M

```

- Iniciar Docker con sistema operativo

sudo systemctl enable --now docker

```

root@JairoBonilla:/home/jairo
[root@JairoBonilla jairo]# sudo systemctl enable --now docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@JairoBonilla jairo]#

```

- Iniciar Docker

systemctl start docker

```

root@JairoBonilla:/home/jairo
[root@JairoBonilla jairo]# systemctl start docker
[root@JairoBonilla jairo]#

```

- Verificar instalación


docker --version

docker compose version

```

root@JairoBonilla:/home/jairo
root@JairoBonilla jairo]# docker --version
Docker version 28.1.1, build 4eba377
root@JairoBonilla jairo]# docker compose version
Docker Compose version v2.35.1
root@JairoBonilla jairo]# S

```

	Instalación aplicaciones en servidor web		Página 16 de 23
	Desarrollo web avanzado		Unidad: 1
	Bonilla Hidalgo Jairo Smith		NRC: 23275

- Ejecutar un hola mundo en el Docker

Primero se crea el archivo html

```
index.html x  dockerfile
index.html > html > body > h1
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Hola Mundo</h1>
10 </body>
11 </html>
```

Crear el Docker file

```
dockerfile x
dockerfile > ...
1  # Usa la imagen oficial de nginx
2  FROM nginx:alpine
3
4  # Copia el HTML a la carpeta que nginx sirve por defecto
5  COPY . /usr/share/nginx/html
6
7  # Expone el puerto 80
8  EXPOSE 80
9
```

Abrir la terminal y construir la imagen en Docker

`docker build -t hola-mundo .`


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
[root@JairoBonilla holaDocker]# docker build -t hola-mundo .
[+] Building 9.0s (7/7) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 282B
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 547B
=> [1/2] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a1
=> => resolve docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a1
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB
=> => sha256:6769dc3a703c719c1d2756bd113659be28ae16cf0da58dd5fd823d6b9a050ea 10.79kB / 10.79kB
=> => sha256:62223d644fa234c3a1cc785ee14242ec47a77364226f1c811d2f669f96dc2ac8 2.50kB / 2.50kB
=> => sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc15fb093229f648674 1.79MB / 1.79MB
=> => sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10 10.33kB / 10.33kB
=> => sha256:b464cfd2a6319875aeb27359ec549790ce14d8214fcb16ef915e4530e5ed235 629B / 629B
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 1.6s
```

Corer la imagen:

`docker run -d -p 80:80 --name contenedor-hola hola-mundo`

```
[root@JairoBonilla holaDocker]# docker run -d -p 80:80 --name contenedor-hola hola-mundo
153689da6fca4e1add734370af4d20c0a9b04e2e94cb18ef6a1f14749daaa6dd
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity
on endpoint contenedor-hola (488dc70b8ec8e1b27fd1301112405b50d7f9380d233bfdb1fc861708f82c73ca): failed to bind host port fo
r 0.0.0.0:80:172.17.0.2:80/tcp: address already in use

Run 'docker run --help' for more information
[root@JairoBonilla holaDocker]#
```


	Instalación aplicaciones en servidor web	Página 17 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

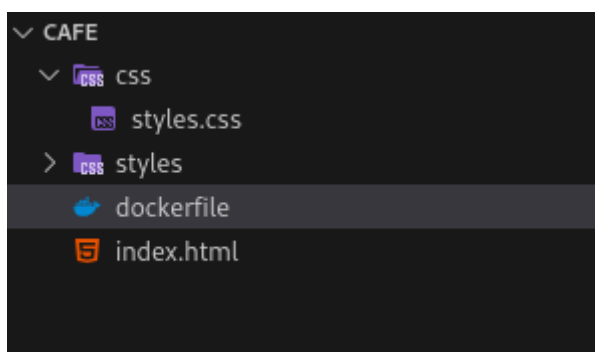
```
Run 'docker run --help' for more information
[root@JairoBonilla holaDocker]# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
153689da6fca   hola-mundo "/docker-entrypoint...." 30 seconds ago Created              contenedor-hola
[root@JairoBonilla holaDocker]#
```

Correr el contenedor con docker start contenedor-hola

```
[root@JairoBonilla holaDocker]# docker start contenedor-hola
Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint contenedor-hola (8b5c3948a81f0adfd80090b6d106adae45878837e68fe2aff69375bd3fc6533b): failed to bind host port for 0.0.0.0:80:172.17.0.2:80/tcp: address already in use
Error: failed to start containers: contenedor-hola
[root@JairoBonilla holaDocker]#
```

7. Ejercicio de dockerizar una web

- Crear una página web de café:




- Crear el archivo dockerfile

```
dockerfile x
dockerfile > ...
1 # Usa la imagen oficial de nginx
2 FROM nginx:alpine
3
4 # Copia el HTML a la carpeta que nginx sirve por defecto
5 COPY . /usr/share/nginx/html
6
7 # Expone el puerto 80
8 EXPOSE 80
9
```

- Construir la imagen docker

docker build -t imagen-pagina-cafe .

	Instalación aplicaciones en servidor web	Página 18 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

```
[root@JairoBonilla cafe]# docker build -t imagen-pagina-cafe .
[+] Building 2.6s (8/8) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 210B
=> [internal] load metadata for docker.io/library/nginx:alpine
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 424B
=> [1/2] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10
=> CACHED [2/2] COPY . /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:913e4d5f4ca594a389aa6f52209780647b884bf8e1a17cb100e673d9489662ce
=> => naming to docker.io/library/imagen-pagina-cafe
[root@JairoBonilla cafe]#
```

- Ejecutar la imagen como un contenedor

```
docker run -d -p 8080:80 --name mi-web-cafe imagen-pagina-cafe
```

```
[root@JairoBonilla cafe]# docker run -d -p 8080:80 --name mi-web-cafe imagen-pagina-cafe
d1d6d725cdcb8813b2904245e2aa33e468e85034067c908189b71cf261eeb295
[root@JairoBonilla cafe]#
```

- Abrir el navegador y acceder a la página




8. Publicar un proyecto en docker hub

- Iniciar sesión desde la terminal

```
docker login
```

```
[root@localhost jairo]# docker login
```

- Acceder al link y poner el código de activación:

	Instalación aplicaciones en servidor web	Página 19 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

USING WEB-BASED LOGIN

Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: **ZZHB-BTRV**

Press **ENTER** to open your browser or submit your device code here: <https://login.docker.com/activate>

Waiting for authentication in the browser...

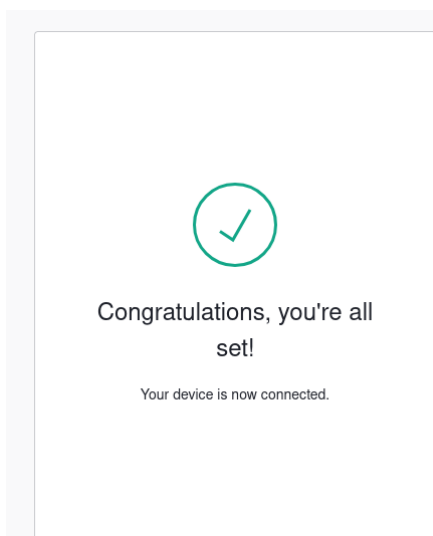
WARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
Configure a credential helper to remove this warning. See
<https://docs.docker.com/go/credential-store/>

- Si todo está bien en la consola indica un mensaje de éxito (Login succeeded)

```
[root@JairoBonilla jairo]# docker login
Authenticating with existing credentials... [Username: jsbonilla2]

Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
[root@JairoBonilla jairo]#
```




- Crear una etiqueta:

docker tag imagen-pagina-cafe jsbonilla2/imagen-pagina-cafe

```
Login Succeeded
[root@JairoBonilla jairo]# docker tag imagen-pagina-cafe jsbonilla2/imagen-pagina-cafe
```

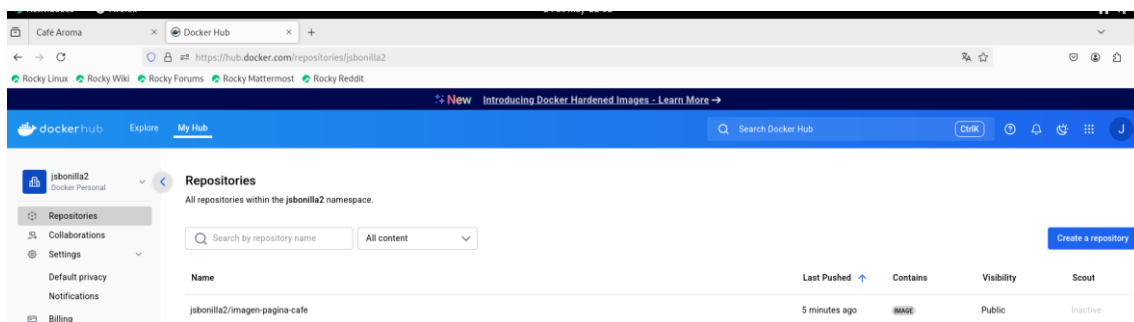
- Publicar la imagen en Docker hub:

docker push jsbonilla2/imagen-pagina-café

	Instalación aplicaciones en servidor web	Página 20 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

```
[root@JairoBonilla jairo]# docker push jsbonilla2/imagen-pagina-cafe
Using default tag: latest
The push refers to repository [docker.io/jsbonilla2/imagen-pagina-cafe]
db0652fb2f03: Mounted from jsbonilla2/dwa
0d853d50b128: Mounted from jsbonilla2/dwa
947e805a4ac7: Mounted from jsbonilla2/dwa
811a4dbbf4a5: Mounted from jsbonilla2/dwa
b8d7d1d22634: Mounted from jsbonilla2/dwa
e244aa659f61: Mounted from jsbonilla2/dwa
c56f134d3805: Mounted from jsbonilla2/dwa
d71eae0084c1: Mounted from jsbonilla2/dwa
08000c18d16d: Mounted from jsbonilla2/dwa
latest: digest: sha256:8df486b657d2fe8ff0b4b2727dea7ed6e9753fd7adbe5cb0ff79722fbd0dc105 size: 2198
```

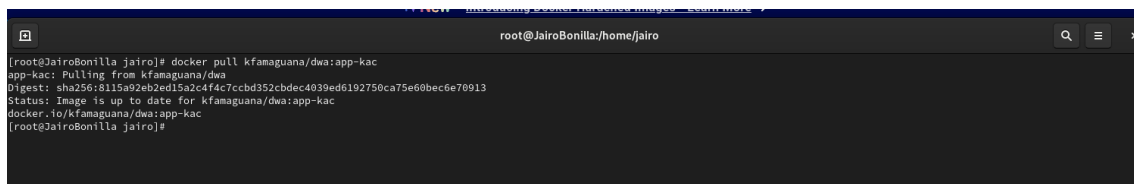
- Verificar en el Docker hub que esté subida la imagen



9. Instalar el docker de un compañero de sus aulas

- Descargar la imagen Docker del compañero

`docker pull kfamaguana/dwa:app-kac`



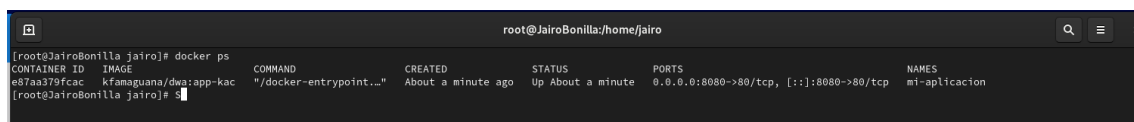
- Ejecutar la imagen como un contenedor


`docker run -d -p 8080:80 --name mi-aplicacion kfamaguana/dwa:app-kac`

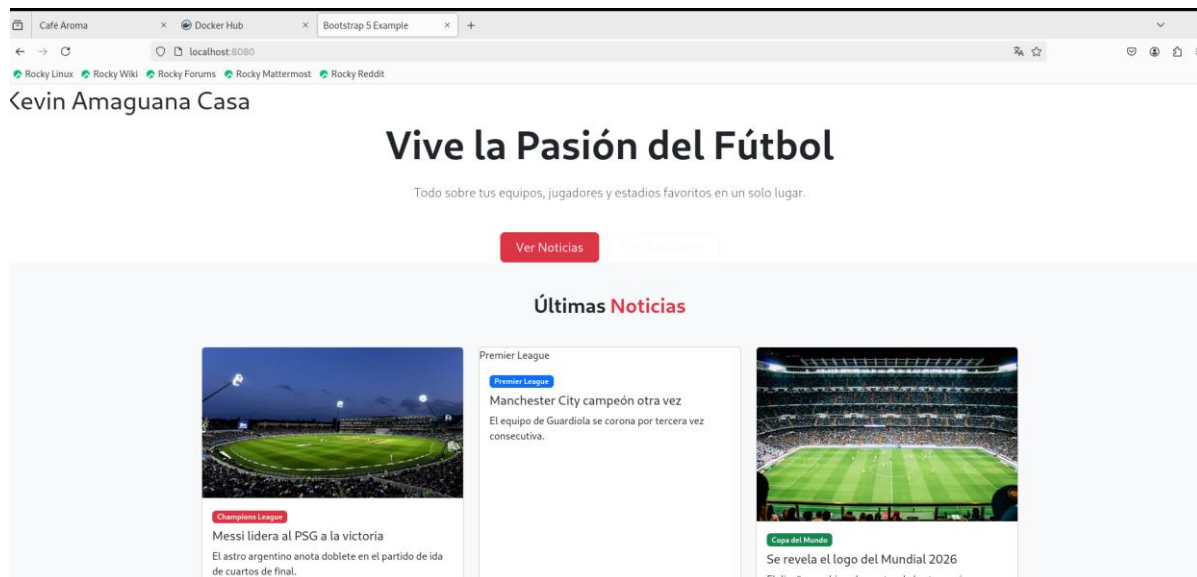
```
[root@JairoBonilla jairo]# docker run -d -p 8080:80 --name mi-aplicacion kfamaguana/dwa:app-kac
e87aa379fcac965107afe2570032e6f6019b3758dfc7f58b8c71d04eb569d7a1
```

- Verificar que el contenedor esté corriendo

`docker ps`



	Instalación aplicaciones en servidor web		Página 21 de 23	
	Desarrollo web avanzado		Unidad:	1
	Bonilla Hidalgo Jairo Smith		NRC:	23275



10. Instalar la base de datos postgres

- Ejecutar el siguiente comando


```
docker run --ulimit memlock=-1:-1 --name postgres -e POSTGRES_USER=sa -e POSTGRES_PASSWORD=admin -e POSTGRES_DB=espe -p 5432:5432 -d postgres
```

```
root@JairoBonilla:/home/jairo
[root@JairoBonilla jairo]# docker run --ulimit memlock=-1:-1 --name postgres -e POSTGRES_USER=sa -e POSTGRES_PASSWORD=admin -e POSTGRES_DB=espe -p 5432:5432 -d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
61320b01ae5e: Pull complete
3db9b37be7c3: Pull complete
e9a82aed48d7: Pull complete
7d852ebdd63e: Pull complete
28708ff4e046: Pull complete
6ce13d85dabe: Pull complete
bd1fa28722bb: Pull complete
410cd7ec9a40: Pull complete
479b0e32b814: Pull complete
e7ab16d6a5e: Pull complete
89ba8b615fa9: Pull complete
82697a7976df: Pull complete
7e11eb1422f3: Pull complete
2bb588ce4e67: Pull complete
Digest: sha256:6efdd0f010dc3cb40d5e33e3ef84acecc5e73161bd3df06029ee8698e5e12c60
Status: Downloaded newer image for postgres:latest
95c670bbf11abb379fa35eb605a9321b35ce2f301646ea43987aabfe448e86a6
[root@JairoBonilla jairo]#
```

- Verificar que esté corriendo correctamente

docker ps

```
root@JairoBonilla:/home/jairo
[root@JairoBonilla jairo]# docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
95c670bbf11a   postgres  "docker-entrypoint.s..." About a minute ago Up About a minute   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp   postgres
e87aa379fcac   kfamaguana/dwa:app-kac "/docker-entrypoint..." 23 minutes ago Up 23 minutes     0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   mi-aplicacion
[root@JairoBonilla jairo]#
```


	Instalación aplicaciones en servidor web	Página 22 de 23	
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

CONCLUSIONES

La realización de estas actividades permitió comprender de manera integral los procesos necesarios para instalar, configurar y administrar servidores web y entornos de contenedores. Se evidenció la importancia de dominar tanto sistemas operativos como Rocky Linux, como herramientas modernas como Docker para facilitar el despliegue de aplicaciones. Asimismo, se fortalecieron habilidades en el manejo de bases de datos. Este conjunto de conocimientos resulta esencial para el trabajo profesional en desarrollo y administración de sistemas.

PARA MI

El conocimiento adquirido me permitirá crear entornos de desarrollo replicables usando Docker, lo cual es útil para proyectos colaborativos. También podré administrar servidores y bases de datos de manera más profesional. Este aprendizaje es aplicable tanto en prácticas académicas como en el despliegue real de sistemas en empresas.

	Instalación aplicaciones en servidor web		Página 23 de 23
	Desarrollo web avanzado	Unidad:	1
	Bonilla Hidalgo Jairo Smith	NRC:	23275

BIBLIOGRAFIA

Docker Inc. (s.f.). *Docker Hub*. Docker. Recuperado el 24 de mayo de 2025, de <https://hub.docker.com/>

Rocky Enterprise Software Foundation. (s.f.). *Rocky Linux*. Recuperado el 24 de mayo de 2025, de <https://rockylinux.org/>

Uniform Server Development Team. (s.f.). *The Uniform Server*. Recuperado el 24 de mayo de 2025, de <https://www.uniformserver.com/>

Elaborado por:	Revisor por:
<p>-----</p> <p>Nombre: Jairo Smith Bonilla Hidalgo Fecha: 26/05/2025</p>	<p>-----</p> <p>Nombre: Fecha:</p>