

Laboratorio 3:

Creación de un Dockerfile, Construcción de una Imagen, Publicación en Docker Hub y Gestión de Contenedores

Paso 1: Preparación (opcional)

Comprobar la instalación de Docker en las máquinas de clase.

```
docker --version
```

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER> docker --version
Docker version 24.0.6, build ed223bc
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER> |
```

Paso 2: Crear un Directorio para el Proyecto

Cada estudiante crea un nuevo directorio para su proyecto.

```
mkdir mi_proyecto_docker
cd mi_proyecto_docker
```

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER> mkdir mi_proyecto_docker

Directorio: C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER

Mode                LastWriteTime         Length Name
----                -
d-----          24/11/2023   17:52             mi_proyecto_docker

PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER> cd mi_proyecto_docker
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> |
```

Paso 3: Crear Archivos HTML

Dentro del directorio del proyecto, cada estudiante crea tres archivos HTML con nombres propios

(por ejemplo, `index_andrei.html`, `index_elena.html`, `index_arturo.html`, opcional incluye imágenes de mascotas favoritas, utiliza un sitio web libre de derechos para descargar una imagen, ejemplo pixabay.com o unsplash.com).

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> New-Item -ItemType File -Name index_carlos2.html

Directorio: C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker

Mode                LastWriteTime         Length Name
----                -
-a----           24/11/2023   18:02             0 index_carlos2.html

PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> New-Item -ItemType File -Name index_carlos3.html

Directorio: C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker

Mode                LastWriteTime         Length Name
----                -
-a----           24/11/2023   18:03             0 index_carlos3.html

PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> |
```

Paso 4: Crear el Dockerfile

En el mismo directorio, cada estudiante crea un archivo llamado `Dockerfile` con el siguiente contenido:

```
```Dockerfile
FROM nginx:latest

COPY index_juan.html /usr/share/nginx/html/index.html
COPY index_maria.html /usr/share/nginx/html/index_maria.html
COPY index_pedro.html /usr/share/nginx/html/index_pedro.html

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```
```

```
C: > Users > carlo > OneDrive > Escritorio > DAM > ENTORNOS > PRACTICAS_DOCKER > mi_proyecto_docker > Dockerfile
1 FROM nginx:latest
2
3 COPY --chown=nginx:nginx index_carlos.html /usr/share/nginx/html/index_carlos.html
4 COPY --chown=nginx:nginx index_carlos2.html /usr/share/nginx/html/index_carlos2.html
5 COPY --chown=nginx:nginx index_carlos3.html /usr/share/nginx/html/index_carlos3.html
6 COPY --chown=nginx:nginx gatito.jpg /usr/share/nginx/html/gatito.jpg
7
8 EXPOSE 80
9
10 CMD ["nginx", "-g", "daemon off;"]
```

Paso 5: Construir la Imagen

Cada estudiante ejecuta el siguiente comando para construir la imagen:

```
```terminal
docker build -t nombre-de-usuario/mi_proyecto_docker:tag .
```
```

El ejemplo de mi terminal de VSC, para este paso:

El comando:

```
docker build -t duscam23/prueba-crear-imagen-andr_docker:0.1
"C:\Users\andri\Desktop\23_24_Benigasl \Entornos de
Desarrollo\Proyecto1DockerfileAnd23"
```

Donde para el comando

1. **`docker build`**: Este es el comando principal de Docker para construir imágenes.
 2. **`-t duscam23/prueba-crear-imagen-andr_docker:0.1`**: Aquí, usas la opción **`-t`** para etiquetar (tag) la imagen que estás construyendo. En este caso, estás asignando el nombre **`duscam23/prueba-crear-imagen-andr_docker`** a la imagen y la etiqueta **`0.1`** para identificar la versión.
- `duscam23`**- mi nombre de usuario del dockerhub

3. **"C:\Users\andri\Desktop\23_24_Benigasl \Entornos de Desarrollo\Proyecto1DockerfileAnd23"** -

Esta es la ruta al contexto de construcción. El contexto es el conjunto de archivos y directorios que Docker enviará al daemon para construir la imagen.

En este caso, estoy especificando la ruta del directorio que contiene el Dockerfile y otros archivos necesarios para construir la imagen. Están en mi carpeta.

Finalmente, con este comando estoy instruyendo a Docker para construir una imagen utilizando el Dockerfile ubicado en el directorio especificado y etiquetarla con el nombre **`duscam23/prueba-crear-imagen-andr_docker`** y la versión **`0.1`**. Este es un paso fundamental en el proceso de construcción de imágenes Docker antes de ejecutar contenedores basados en esas imágenes.

Por ejemplo, si este Dockerfile se usa para construir una imagen y luego se ejecuta un contenedor basado en esa imagen, el contenedor iniciará Nginx con la configuración especificada por el comando **`CMD`**.

Además

El comando **`CMD`** en un Dockerfile se utiliza para especificar el comando predeterminado que se ejecutará cuando se inicie un contenedor basado en la imagen. La forma básica del comando **`CMD`** es:

```
'''archivo dockerfile
CMD [comando, argumento1, argumento2, ...]
'''
```

En mi caso específico, el comando `CMD` está configurado para iniciar el servidor Nginx en el contenedor con los argumentos `"-g daemon off;"`. Desglosemos cada parte del comando:

1. `**`nginx`:**` Este es el comando principal que se ejecutará en el contenedor. En este caso, se está iniciando el servidor web Nginx.
2. `**`-g`:**` Este es un argumento para el comando Nginx. En este contexto, se utiliza para proporcionar una directiva de configuración al servidor.
3. `**`daemon off;`:**` Esta es la directiva de configuración específica que se pasa al servidor Nginx. En este caso, se establece ``daemon`` en ``off``, lo que significa que el servidor Nginx se ejecutará en el primer plano (foreground) en lugar de en el fondo (background). Esto es común en entornos de contenedor donde los procesos deben ejecutarse en el primer plano para que el contenedor no salga cuando el proceso principal finaliza.

MI respuesta:

[illegible]

Importante: asegúrate que tu archivo dockerfile no tiene extensión:

Si quieres cambiar la extensión del archivo "Untitled-1.dockerfile" a "Dockerfile". Puedes hacerlo utilizando el comando `Rename-Item` en PowerShell. Aquí está el comando que puedes usar:

Terminal visual studio code

```
Rename-Item -Path .\Untitled-1.dockerfile -NewName Dockerfile
```

Este comando cambiará el nombre del archivo "Untitled-1.dockerfile" a "Dockerfile"

Paso 6: Iniciar Sesión en Docker Hub

Cada estudiante inicia sesión en Docker Hub desde la línea de comandos:

```terminal de visual estudio code:

docker login

```

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> docker login
Authenticating with existing credentials...
Login Succeeded
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> |
```

Paso 7: Subir la Imagen a Docker Hub

Cada alumno/a sube su imagen al repositorio en Docker Hub:

```terminal de visual estudio code:

Paso opcional, comprueba las imágenes:

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
carperraj/mi_proyecto_docker 0.1 ede9ac5c27cf About a minute ago 187MB
hello-world latest 9c7a54a9a43c 6 months ago 13.3kB
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> |
```

docker push nombre-de-usuario/mi\_proyecto\_docker:tag

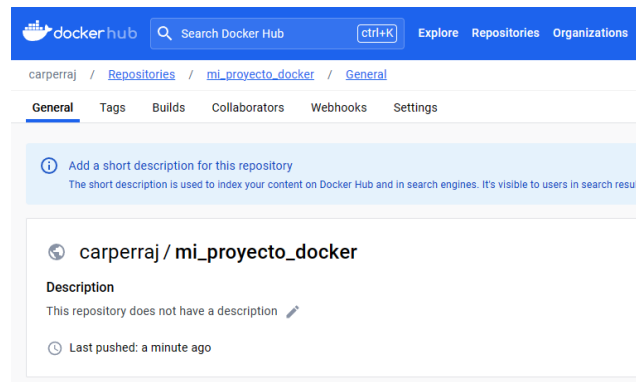
```

docker push duscarn23/prueba-crear-imagen-andr_docker:0.1

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> docker push carperraj/mi_proyecto_docker:0.1
The push refers to repository [docker.io/carperraj/mi_proyecto_docker]
bba49b6fa05d: Pushed
cb2869997af2: Pushed
b44fde9bac72: Pushed
d5520be953a2: Pushed
0d0e9c83b6f7: Layer already exists
cddc309885a2: Layer already exists
c2d3ab485d1b: Layer already exists
66283570f41b: Layer already exists
f5525891d9e9: Layer already exists
8ae474e0cc8f: Layer already exists
92770f546e06: Layer already exists
0.1: digest: sha256:5a3411ab188ed8a2591086827c0ee3fc87806258cd0c84ee25153cf7da91966b size: 2609
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> |
```

Comprobar en dockerHub si esta tu imagen.

Veo que tengo un proyecto: **prueba-crear-imagen-andr_docker**
y sí le damos clic:



Vamos a utilizar este comando para crear contenedores y realizar operaciones de volúmenes y de mapeado:

```
docker pull carperraj/mi_proyect...
```

Paso 8: Iniciar Contenedores

Cada estudiante inicia 4-5 contenedores usando su imagen:

```terminal

```
docker run -d -p 8081:80 nombre-de-usuario/mi_proyecto_docker:tag
docker run -d -p 8082:80 nombre-de-usuario/mi_proyecto_docker:tag
... continuar según sea necesario
```

```
docker run -d -p 8081:80 duscarn23/prueba-crear-imagen-andr_docker:0.1
docker run -d -p 8082:80 duscarn23/prueba-crear-imagen-andr_docker:0.1
docker run -d -p 3000:80 --name prueba-1-docker duscarn23/prueba-crear-
imagen-andr_docker:0.1
```

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS DOCKER\mi_proyecto_docker> docker run -d -p 8081:80 carperraj/mi_proyecto_docker:0.1
88b2e328c42a34b33a2ff9c20bfcabe3cf1241ca746e759132decc6e3811d11b
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS DOCKER\mi_proyecto_docker> docker run -d -p 8082:80 carperraj/mi_proyecto_docker:0.1
6d6c40e3a47c5f7c7a8781acdd8d8ab8c94367a1b51e74091efe715b801ac93
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS DOCKER\mi_proyecto_docker> docker run -d -p 3000:80 --name prueba-1-docker carperraj/mi_proyecto_docker:0.1
b41669587783ec338f321c569e805f515b735369e137eb0fcae01a70d97095ab
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS DOCKER\mi_proyecto_docker>
```

## Paso 9: Verificar Contenedores en Ejecución

Cada estudiante verifica que los contenedores están en ejecución:

```terminal

docker ps -a

```
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
b41669587783   carpennaj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 42 seconds ago Up 41 seconds 0.0.0.0:3000->80/tcp               prueba-1-docker
6d6c4a0e3a47   carpennaj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 2 minutes ago  Up 2 minutes  0.0.0.0:8082->80/tcp               pensive_merkle
88b2e328c42a   carpennaj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 2 minutes ago  Up 2 minutes  0.0.0.0:8081->80/tcp               awesome_beaver
8446c8313634   hello-world                         "/hello"                 36 minutes ago Exited (0) 36 minutes ago         sad_chandrasekhar
PS C:\Users\carlo\OneDrive\Escritorio\DAW\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker>
```

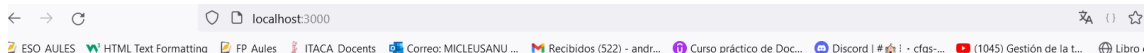
```

Está es la distribución es decir el mapeo de puertos para cada contenedor:

- prueba-1-docker: Mapea el puerto 3000 de tu máquina host al puerto 80 del contenedor.
- pensive\_merkle: Mapea el puerto 8082 de tu máquina host al puerto 80 del contenedor.
- awesome\_beaver: Mapea el puerto 8081 de tu máquina host al puerto 80 del contenedor.

Esto significa *que, por ejemplo, si accedes a <http://localhost:3000> desde tu navegador, deberías poder ver el contenido del contenedor prueba-1-docker.*

Comprobamos en nuestro pc en localhost:3000



**¡Bienvenido a la Mascota Dockerizada 1!**

Gatito Dockerizado

## #### Paso 10: Personalizar Contenido Visual

Cada estudiante personaliza el contenido visual de sus páginas HTML para hacer la práctica más representativa.

- a) incluir texto
- b) incluir imágenes

## #### Paso 11: Gestionar Contenedores

- Cambiar el nombre de un contenedor:

```terminal

docker rename nombre-actual nuevo-nombre

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker rename prueba-1-docker pruebasdocker
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
b41669587783   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 21 minutes ago Exited (0) 2 minutes ago           pruebasdocker
6d6c4a0e3a47   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 22 minutes ago Up 22 minutes   0.0.0.0:8082->80/tcp   pensive_merkle
88b2e328c42a   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 23 minutes ago Up 23 minutes   0.0.0.0:8081->80/tcp   awesome_beaver
8446c8313634   hello-world                         "/hello"                 57 minutes ago Exited (0) 57 minutes ago           sad_chandrasekhar
```

```

- Detener un contenedor:

```terminal

docker stop nombre-del-contenedor

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker stop prueba-1-docker
prueba-1-docker
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
b41669587783   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 18 minutes ago Exited (0) 7 seconds ago           prueba-1-docker
6d6c4a0e3a47   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 19 minutes ago Up 19 minutes   0.0.0.0:8082->80/tcp   pensive_merkle
88b2e328c42a   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 20 minutes ago Up 20 minutes   0.0.0.0:8081->80/tcp   awesome_beaver
8446c8313634   hello-world                         "/hello"                 54 minutes ago Exited (0) 54 minutes ago           sad_chandrasekhar
```

```

- Eliminar un contenedor:

```terminal

docker rm nombre-del-contenedor

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker rm pruebasdocker
pruebasdocker
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                    NAMES
6d6c4a0e3a47   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 23 minutes ago Up 23 minutes   0.0.0.0:8082->80/tcp   pensive_merkle
88b2e328c42a   carperraj/mi_proyecto_docker:0.1   "/docker-entrypoint..." 24 minutes ago Up 24 minutes   0.0.0.0:8081->80/tcp   awesome_beaver
8446c8313634   hello-world                         "/hello"                 58 minutes ago Exited (0) 58 minutes ago           sad_chandrasekhar
```

```

## Paso 12: Gestionar Imágenes

- Listar imágenes en la máquina:

```terminal

docker images

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
carperraj/mi_proyecto_docker  0.1        ede9ac5c27cf 33 minutes ago 187MB
hello-world         latest     9c7a54a9a43c 6 months ago  13.3kB
```

PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> █

```

- Eliminar una imagen:

```terminal

docker rmi nombre-de-usuario/mi_proyecto_docker:tag

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> docker rmi carperraj/mi_proyecto_docker:0.1
Untagged: carperraj/mi_proyecto_docker:0.1
Untagged: carperraj/mi_proyecto_docker@sha256:5a3411ab188ed8a2591086827c0ee3fc87806258cd0c84ee25153cf7da91966b
Deleted: sha256:ede9ac5c27cfc948530311ba436aaba9c94edab808c930f4f67cfb63e106ec4
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRATICAS_DOCKER\mi_proyecto_docker> █
```



```

- Eliminar todas las imágenes no utilizadas:

``` terminal

docker image prune -a

```
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> docker image prune -a
WARNING! This will remove all images without at least one container associated to them.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
PS C:\Users\carlo\OneDrive\Escritorio\DAM\ENTORNOS\PRACTICAS_DOCKER\mi_proyecto_docker> |
```

Nnnnnn

.