

## Laboratorio 4

2.

Después de clonar el repositorio brindado por la gente de Docker, estando dentro de la carpeta clonada, y, utilizando una terminal en esa ubicación, se ingresa el comando *New-Item -Path . -Name Dockerfile -ItemType File* (en el Powershell de Windows) se crea un archivo Dockerfile vacío.

Después de colocar el siguiente contenido al Dockerfile:

```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

Los comandos anteriormente utilizados se encargan de:

- El FROM de instalar y establecer la imagen base para construir la imagen de Docker, proporcionando el entorno y herramientas necesarias para ejecutar la aplicación.
- El WORKDIR establece el directorio de trabajo dentro del contenedor como /app, todos los comandos posteriores se ejecutarán en este directorio.
- El COPY todos los archivos y directorios desde el contexto de construcción (el directorio donde se encuentra el Dockerfile), al directorio de trabajo dentro del contenedor. En este caso el primer punto se refiere al contexto de construcción, y el segundo al directorio de trabajo.
- El RUN ejecuta el comando dentro del contenedor.
- EL CMD define el comando predeterminado que se ejecutará cuando se inicie un contenedor.
- Expone el puerto 3000 en el contenedor.

Y al ejecutar el comando *docker build -t getting-started .*, se utiliza el archivo para crear una nueva imagen con el nombre getting-started, esto a través del -t, y usando el . para decirle al Daemon que busque el archivo Dockerfile en el directorio.

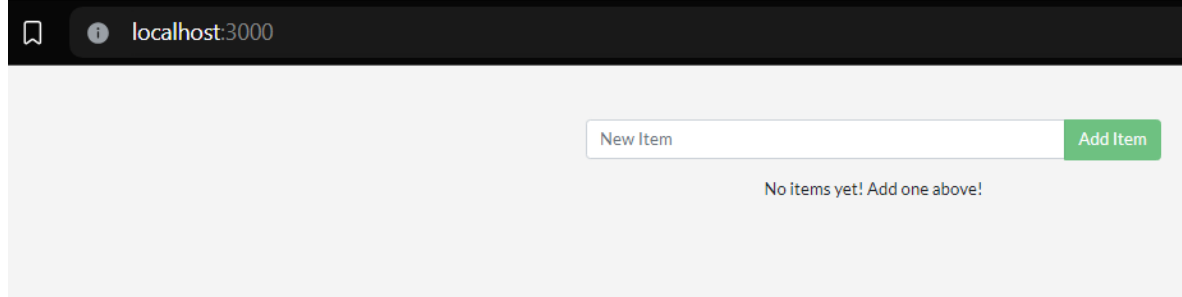
El comando *docker run -dp 127.0.0.1:3000:3000 getting-started* inicializa el contenedor, el comando -d lo despliega en segundo plano, y el -p crea un mapeo de puerto entre el host y el contenedor, en este caso siendo el host el localhost, puerto 3000 del host, con en el puerto 3000 del contenedor.

Lo siguiente es la captura del comando anterior, y el comando `docker ps`, que se encarga de mostrar los contenedores en ejecución:

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-started
658cb064fd6ae1c3a955546dd9530d8b65151dd6a8537e9c1e5a8b69b61eda58
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
658cb064fd6a	getting-started	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	127.0.0.1:3000->3000/tcp	gallant_tu

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app>
```



3.

Después de realizar el cambio en el código, y de montar el contenedor, al tratar de inicializarlo sale el mensaje:

```
docker: Error response from daemon: driver failed programming external connectivity on endpoint jolly_wing (570af13793b68f2674c3c652140941c46459223a4e6970e15ce85564238b827e): Bind for 127.0.0.1:3000 failed: port is already allocated.
```

Esto ocurre porque el anterior contenedor todavía sigue utilizando el puerto 3000 del host, por lo cual hay que removerlo del puerto. Esa acción se puede hacer al ejecutar los comandos `docker stop <contenedor-id>` y `docker rm <contendor-id>`, el primero es para detener la ejecución del contenedor anterior, y el siguiente es el que se encargará de eliminar

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
658cb064fd6a	1be2050f3191	"docker-entrypoint.s..."	51 minutes ago	Up 51 minutes	127.0.0.1:3000->3000/tcp	gallant_tu

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker stop 658cb064fd6a
658cb064fd6a
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker ps
```

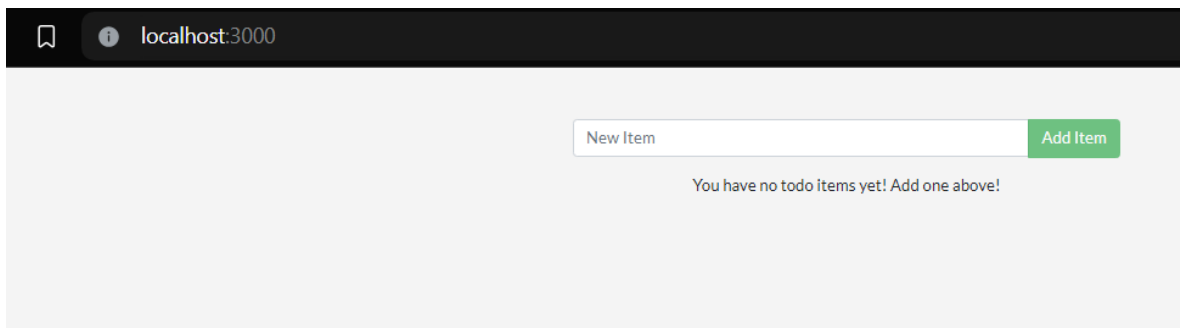
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker rm 658cb064fd6a
658cb064fd6a
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-started
9f28a3fd6a99a90d10d454b692065f5df9ac170fef20783628e66024f8336bf2
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9f28a3fd6a99	getting-started	"docker-entrypoint.s..."	About a minute ago	Up About a minute	127.0.0.1:3000->3000/tcp	musling williams



4.

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS                    NAMES
03d8625a4439   getting-started     "docker-entrypoint.s..." About a minute ago Up About a minute   127.0.0.1:3000->3000/tcp   beautiful-kepler
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker stop 03d8625a4439
03d8625a4439
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker rm 03d8625a4439
03d8625a4439
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker run -dp 127.0.0.1:3000:3000 --mount type
=volume,src=todo-db,target=/etc/todos getting-started
d9d3e3b3402558e215c4fe660d693319381ceabebfb359af31060085337d8bae
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker volume inspect todo-db
[
  {
    "CreatedAt": "2024-06-08T19:47:49Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/todo-db/_data",
    "Name": "todo-db",
    "Options": null,
    "Scope": "local"
  }
]
```

5.

```
mysql> select * from todo_items;
+-----+-----+-----+
| id                | name        | completed |
+-----+-----+-----+
| 242ef360-f802-48aa-bdc6-3c55187cd6dc | frescoleche | 0         |
| d7b22224-b5ba-4ced-9d15-319fd7e69cd8 | pepsi       | 0         |
| 8187b9a6-16aa-4699-8965-55ae1d98ca52 | imperial    | 0         |
| c06ab0d8-f48f-4f64-91c7-14671d79c0e1 | pilsen      | 0         |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> exit
Bye
```

7.

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4> docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
f3b4140d3e3d    bridge   bridge  local
4e65d0a13677    host     host    local
ff54d904a768    none     null    local
ce5af03a723c    todo-app bridge  local
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4> docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
696cfb1a7d61   node:18-alpine "docker-entrypoint.s..." 4 minutes ago  Up 4 minutes  127.0.0.1:3000->3000/tcp            priceless_shamir
f8394a40421c   nicolaka/netshoot "zsh"                  About an hour ago  Up About an hour                               brave_chaum
a371096fd9d9   nicolaka/netshoot "zsh"                  2 hours ago      Up 2 hours                               clever_elgamal
792280e391cf   mysql:8.0    "docker-entrypoint.s..." 2 hours ago      Up 2 hours      3306/tcp, 33060/tcp                busy_bohr
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4>
```

8.

```
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker compose up -d
[+] Running 3/3
 ✓ Network getting-started-app_default      Created
 ✓ Container getting-started-app-app-1      Started
 ✓ Container getting-started-app-mysql-1    Started
PS C:\Users\XPC\Documents\Biblioteca de todo\Universidad Nacional\Año III Ciclo I\Redes de Computadores\Lab4\getting-started-app> docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
4b6db7a85a09   mysql:8.0    "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes  3306/tcp, 33060/tcp                getting-started-app-mysql-1
a14f108aecd2   node:18-alpine "docker-entrypoint.s..." 6 minutes ago  Up 6 minutes  127.0.0.1:3000->3000/tcp            getting-started-app-app-1
```

ENLACE AL REPOSITORIO: <https://github.com/altabored/Laboratorio-4>

En el archivo .gitignore fue omitido el directorio getting-started-app/node\_modules, ya que generaba errores al ser subido a Github.