



**Universidad de Guadalajara.**  
**Centro Universitario de Ciencias Exactas e Ingenierías.**



## **Ejercicio 8**

### **Docker**

### **Computación tolerante a fallas NRC 179961**

**Nombre del alumno:** Loredó Padilla Orlando Javier

**Código:** 217560328

**Departamento:** Departamento de ciencias computacionales

**Nombre del maestro:** López Franco Michel Emanuel

**Carrera:** Ingeniería en computación

**Sección:** D06

**Ciclo:** 23B

**Fecha:** 23/10/2023.

## Introducción

Utiliza Docker para generar una aplicación y si es de microservicios mejor.

## Programa

Para la práctica intentamos crear un contenedor usando Docker. Lo que hicimos fue descargar el programa de Docker para usar sus comandos dentro del sistema, este debe estar abierto durante la ejecución de los comandos. Lo primero fue descargar los sistemas que íbamos a utilizar en este caso fue node y nginx en la versión 1.23.

```
cmd Símbolo del sistema - docker pull nginx:1.23
Microsoft Windows [Versión 10.0.19045.3570]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\orlan> docker pull node
Using default tag: latest
latest: Pulling from library/node
0a9573503463: Pull complete
1ccc26d841b4: Pull complete
800d84653581: Pull complete
7c632e57ea62: Pull complete
cfd71c20c2b4: Pull complete
2a9e0f9fe127: Pull complete
84290778bab6: Pull complete
25e31907d19f: Pull complete
Digest: sha256:bf718fc580177cd927173c8617cf7f527a1b7f62c7de882ee17a42d065f4b70e
Status: Downloaded newer image for node:latest
docker.io/library/node:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview node

C:\Users\orlan> docker pull nginx:1.23
1.23: Pulling from library/nginx
f03b40093957: Downloading [=====] 11.45MB/31.4MB
a85095acb896: Download complete
d24b987aa74e: Download complete
6c1a86118ade: Waiting
9989f7b33228: Waiting
```

Una habiendo descargado las imágenes pasaríamos a mostrar que se descargó y pasaríamos a correr la imagen, luego haríamos un log con la id del contenedor para seguido correrla con un local host.

```
CS: Símbolo del sistema

C:\Users\orlan> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
a2635967f8bc   nginx:1.23 "/docker-entrypoint..." 39 seconds ago Up 20 seconds 80/tcp        thirsty_joliot

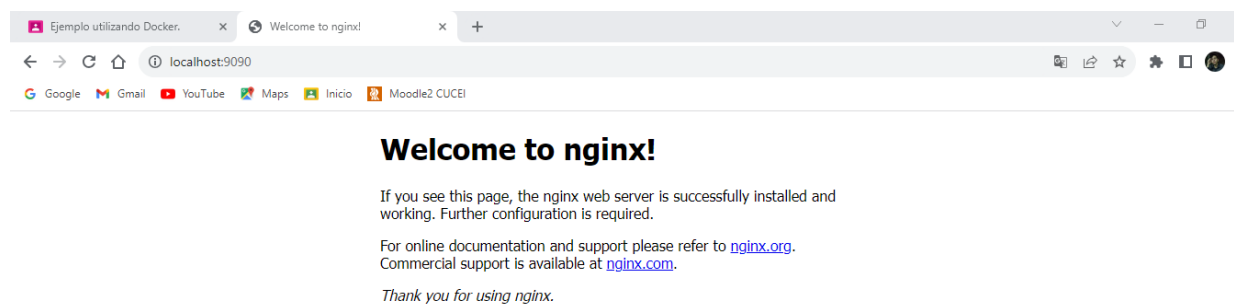
C:\Users\orlan> docker run -d nginx:1.23
4545abb640bc3f8782f1bdb7f3e011d4672675810ab283f1755d72d6b5dda659

C:\Users\orlan> docker logs a2635967f8bc
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/10/20 21:55:06 [notice] 1#1: using the "epoll" event method
2023/10/20 21:55:06 [notice] 1#1: nginx/1.23.4
2023/10/20 21:55:06 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/10/20 21:55:06 [notice] 1#1: OS: Linux 5.15.90.1-microsoft-standard-WSL2
2023/10/20 21:55:06 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/10/20 21:55:06 [notice] 1#1: start worker processes
2023/10/20 21:55:06 [notice] 1#1: start worker process 30
2023/10/20 21:55:06 [notice] 1#1: start worker process 31

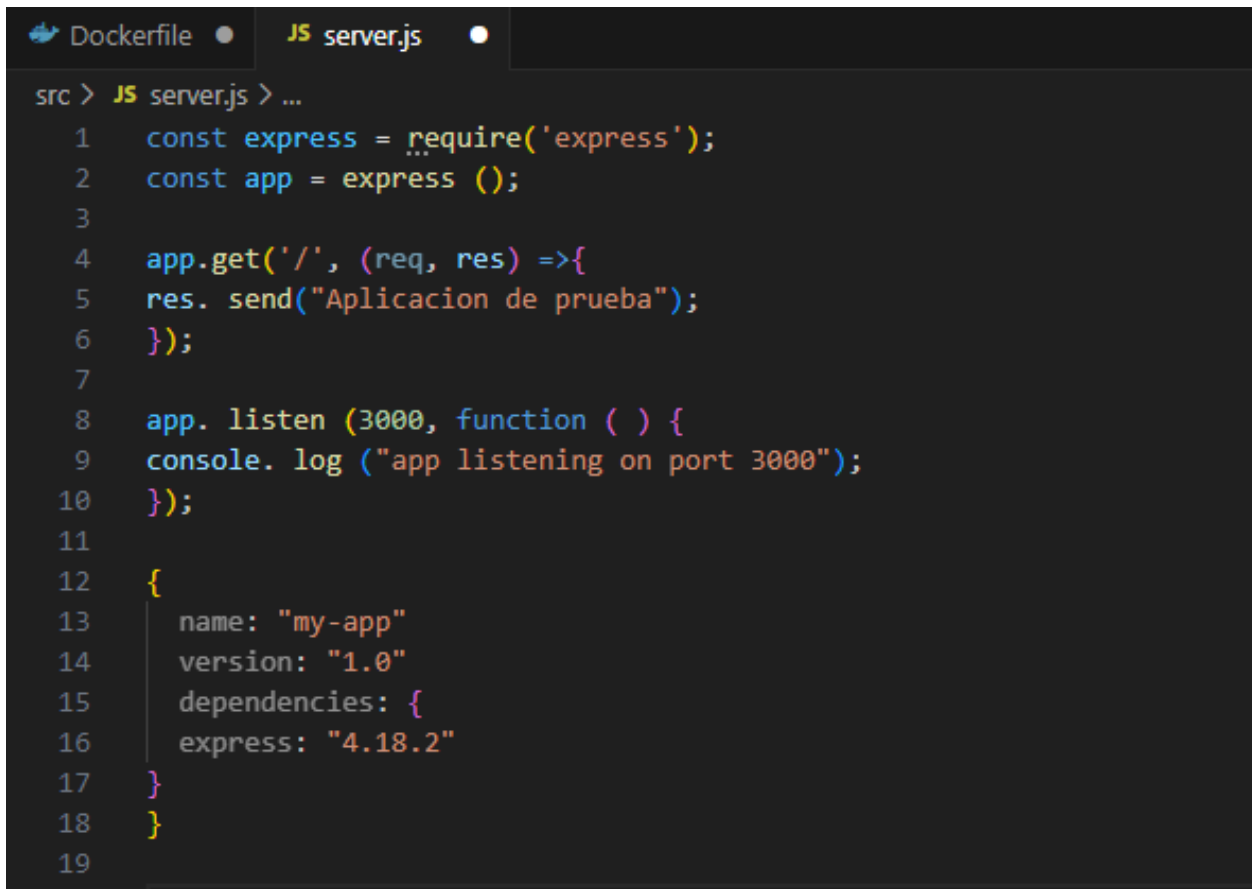
C:\Users\orlan> docker run -d -p 9090:80 nginx:1.23
49059c7535017e5ab5455ab5d8f18534fdaf8e46df2aa3a666f20bc6ccabb5c2

C:\Users\orlan>
```

Al correr la imagen tendremos que ir a Google y buscar localhost con los puertos indicados en este caso fue 9090 y una vez ahí ya tendremos nuestro contenedor activo.



Se intentó ejecutar el siguiente programa para poder modificar la página que se había creado pero el comando de node jamás respondió por lo cual solo se pudo llegar a la parte de crear un contenedor.



```
Dockerfile JS server.js
src > JS server.js > ...
1  const express = require('express');
2  const app = express ();
3
4  app.get('/', (req, res) =>{
5    res. send("Aplicacion de prueba");
6  });
7
8  app. listen (3000, function ( ) {
9    console. log ("app listening on port 3000");
10 });
11
12 {
13   name: "my-app"
14   version: "1.0"
15   dependencies: {
16     express: "4.18.2"
17   }
18 }
19
```

## Conclusión

Docker es un sistema con la misma función que una maquina virtual, con este podremos simular un equipo desde nuestra consola, esto nos permite hacer uso de otros servicios y aprovechar al máximo los recursos de nuestro equipo, además que en este modo el proceso de instalar y configurar un sistema operativo nuevo se omite por completo haciéndolo un método muy cómodo de usar.

Para la práctica intentamos crear una instancia utilizando Docker, aunque esta no pudo llegar a ser más compleja por algunos problemas en el sistema.

## Referencias

- HolaMundo (2022). Aprende Docker ahora! curso completo gratis desde cero!. Recuperado el 18/10/2023 en: <https://youtu.be/4Dko5W96WHg?si=rq-slhaXuYqZCkMF>