UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

Materia: Computación tolerante a fallas

Clave: I7036

Sección: D06

NRC: 179961

Microservicios
Código estudiante: 221350567

Alumno: Rauf Alfonso Hamden Estrada

Carrera: Ingeniería en Computación

Fecha: 12/11/2023

Docente: Michel Emanuel Lopez Franco

2023B

Introducción

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución. Con Docker, se puede implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno

Desarrollo

Cargamos la imagen

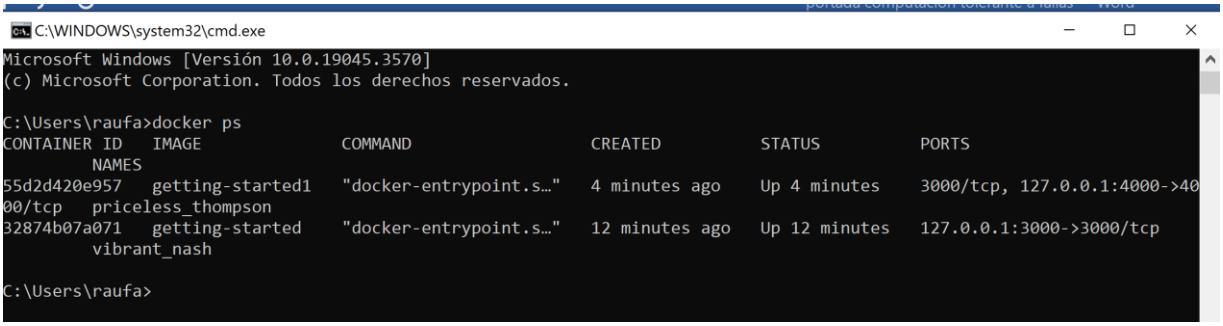
```
ocker: Error response from daemon: driver failed programming external connectivity on endpoint pedantic_leakey (ecee83df336cb
4be278c73d0a): Bind for 127.0.0.1:3000 failed: port is already allocated.
S C:\Users\raufa\OneDrive\Escritorio\Docker\getting-started-app> docker run -dp 127.0.0.1:3000:4000 getting-started1
5d372b293d1ec52ee3e42a0b0facf4d19d3ae594da68984db9022354afcafe6
ocker: Error response from daemon: driver failed programming external connectivity on endpoint wonderful_bartik (931d33a85a7e
c22121a1bc2f5): Bind for 127.0.0.1:3000 failed: port is already allocated.
S C:\Users\raufa\OneDrive\Escritorio\Docker\getting-started-app> docker run -dp 127.0.0.1:4000:4000 getting-started1
5d2d420e95729c8539f8f6bf4b586b1829f18566e30dc514e31eaf9417a5f23
S C:\Users\raufa\OneDrive\Escritorio\Docker\getting-started-app> docker build -t getting-started2 .
+] Building 1.7s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 188B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> resolve image config for docker.io/docker/dockerfile:1
=> CACHED docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab202122f8f50032edf31be002
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:435dcad253bb5b7f347ebc69c8cc52de7c912eb7241098b920f2fc2d7843183d
=> [internal] load build context
=> => transferring context: 5.02kB
=> CACHED [2/4] WORKDIR /app
=> CACHED [3/4] COPY . .
=> CACHED [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> => writing image sha256:66820f446c9aa8f7dcc78bb9eaef21e32823f5f9ce35a8c5c3e01906b4f1fcd7
=> => naming to docker.io/library/getting-started2

hat's Next?
 View a summary of image vulnerabilities and recommendations → docker scout quickview
S C:\Users\raufa\OneDrive\Escritorio\Docker\getting-started-app> docker build -t getting-started1 .
```

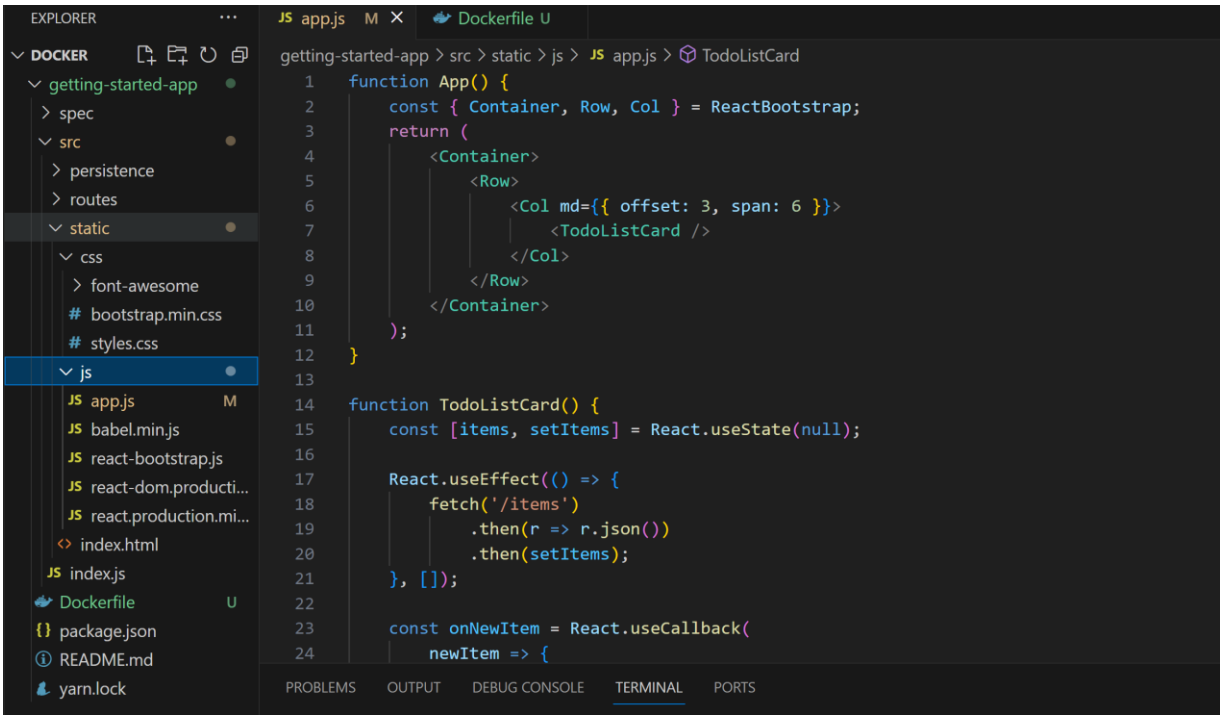
Los inicializamos

```
PS C:\Users\raufa\OneDrive\Escritorio\Docker\getting-started-app> docker run -dp 127.0.0.1:3000:3000 getting-start
c379a378b41c61e6b94d0c8373e0a44ea500414f99b32c397e86a27f55de8701
PS C:\Users\raufa\OneDrive\Escritorio\Docker\getting-started-app> docker build -t getting-started .
[+] Building 25.2s (11/11) FINISHED
```

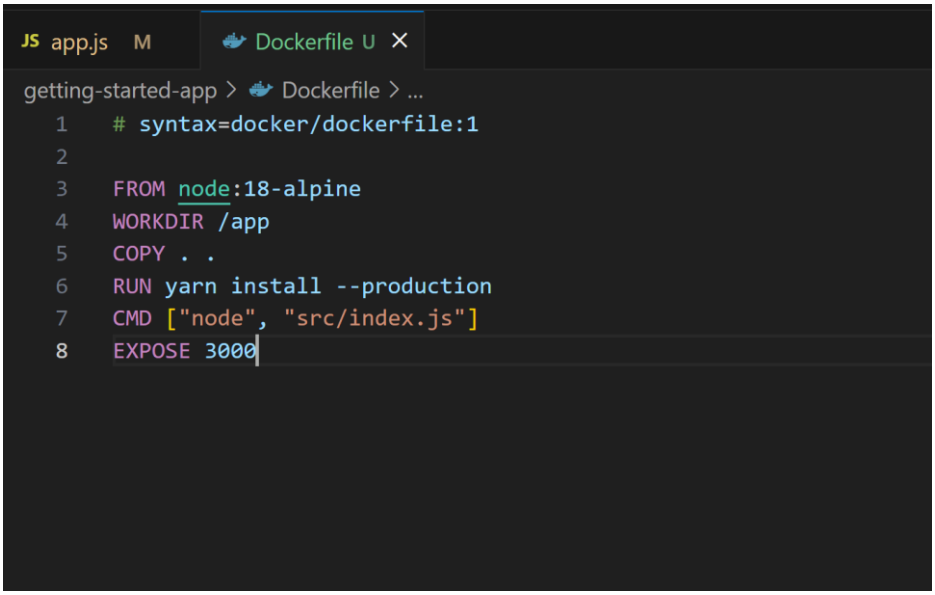
Imágenes creadas



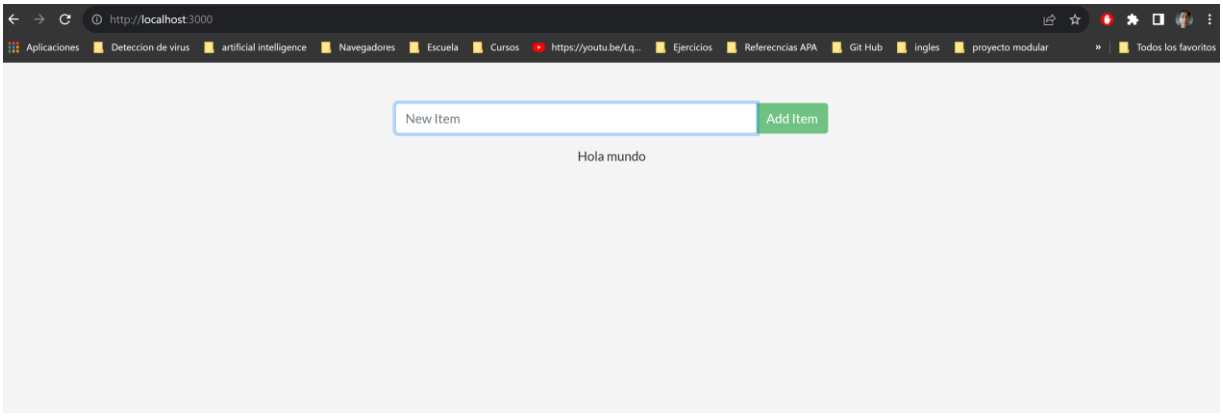
Código

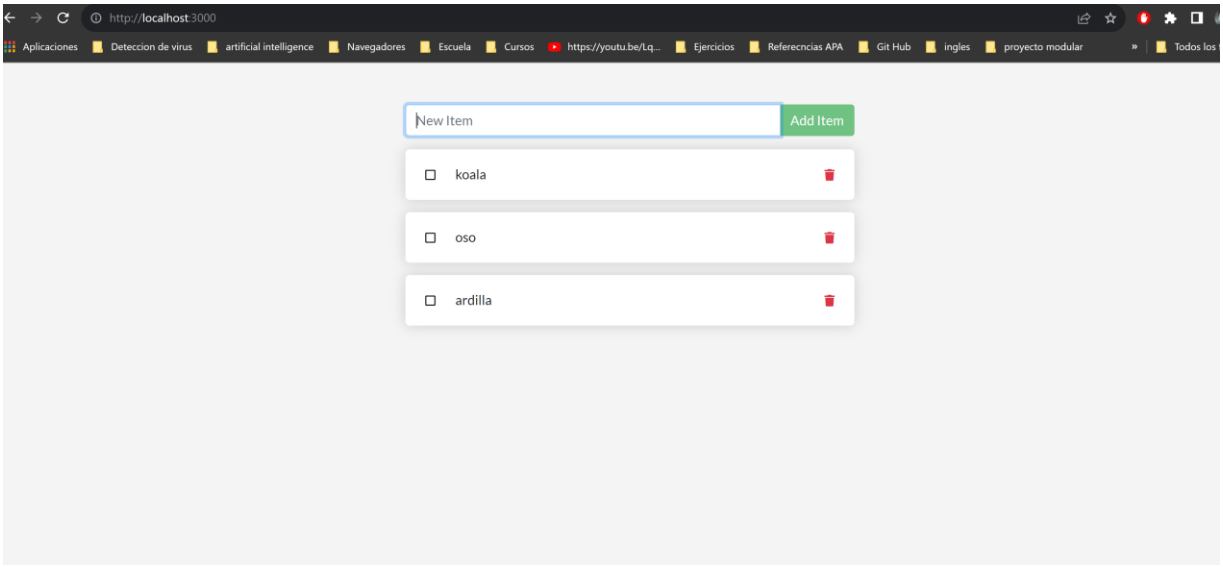


Dockfile



Aplicación en el puerto 3000





Conclusión

Docker al principio era un poco difícil de usar , sin embargo, ya he aprendido como funciona y he podido realizar la practica de micro servicios con la creación de los contenedores. Lo bueno de Docker es que con el podre correr mi código en diferentes maquinas o poder compartirlo sin la preocupación de que este en algún momento me llegue a fallas