

PARTE 1 – Investigación

1. ¿Qué es Docker y para qué sirve?

Docker es una plataforma que permite desarrollar, empaquetar y ejecutar aplicaciones dentro de contenedores, los cuales son entornos ligeros, portables y consistentes que incluyen todo lo necesario para que una app funcione (código, dependencias, configuración, etc.).

2. ¿Qué es un contenedor y en qué se diferencia de una máquina virtual?

Un **contenedor** es una instancia ejecutable de una imagen de Docker. Comparte el sistema operativo del host pero está aislado.

Una **máquina virtual** (VM) emula todo un sistema operativo, incluyendo kernel, lo que la hace más pesada.

3. ¿Qué es una imagen en Docker?

Una imagen es una plantilla de solo lectura que contiene el sistema de archivos, las dependencias, el código y las instrucciones para ejecutar una app.

Es lo que se usa para crear contenedores.

4. ¿Qué es Docker Desktop? ¿Cómo se instala en Windows?

Docker Desktop es la aplicación oficial de Docker para Windows y macOS, que proporciona una interfaz gráfica y todo el entorno necesario (Docker Engine, Docker CLI, etc.).

Instalación en Windows:

- Descargar desde <https://www.docker.com/products/docker-desktop>
- Ejecutar el instalador
- Seguir los pasos y habilitar WSL 2 si lo requiere
- Reiniciar y comenzar a usar

5. ¿Qué es Docker Hub y cómo funciona?

Docker Hub es un repositorio público (como GitHub) donde puedes subir y descargar imágenes Docker.

Funcionamiento:

- `docker pull imagen`: descarga una imagen
- `docker push imagen`: sube una imagen
- También permite crear cuentas, repositorios privados y automatizaciones.

6. ¿Qué son las redes en Docker? Tipos y ejemplos.

Docker permite que los contenedores se comuniquen entre sí usando redes virtuales.

Tipos principales:

- `bridge` (por defecto): red aislada para comunicación entre contenedores
- `host`: usa la red del host directamente
- `none`: sin red

Ejemplo:

- `docker network create mi_red`
- `docker run -d --name app1 --network mi_red nginx`
- `docker run -d --name app2 --network mi_red alpine`

7. Comandos esenciales de Docker:

- a. docker build: crea una imagen a partir de un Dockerfile

```
docker build -t mi-imagen .
```

- b. docker run: ejecuta un contenedor

```
docker run -d -p 3000:3000 mi-imagen
```

- c. docker ps: lista los contenedores en ejecución

- d. docker stop: detiene un contenedor

```
docker stop nombre_o_id
```

- e. docker rm: elimina un contenedor

```
docker rm nombre_o_id
```

- f. docker rmi: elimina una imagen

```
docker rmi nombre_o_id
```

- g. docker images: lista las imágenes locales

- h. docker exec: ejecuta un comando dentro de un contenedor activo

```
docker exec -it nombre bash
```

8. ¿Qué es Dockerfile? ¿Para qué sirve?

Es un archivo de texto que contiene instrucciones para construir una imagen Docker.

Sirve para automatizar el proceso de creación de imágenes, incluyendo instalación de paquetes, copiado de archivos, ejecución de comandos, etc.

9. ¿Qué es .dockerignore?

Es un archivo similar a .gitignore, usado para excluir archivos o carpetas al construir una imagen.

10. ¿Qué es vite preview y cómo se usa en producción?

vite preview es el comando de Vite que sirve para previsualizar el build de producción localmente.

```
npm run build
```

```
npm run preview
```

11. ¿Qué es Kubernetes y cómo se relaciona con Docker?

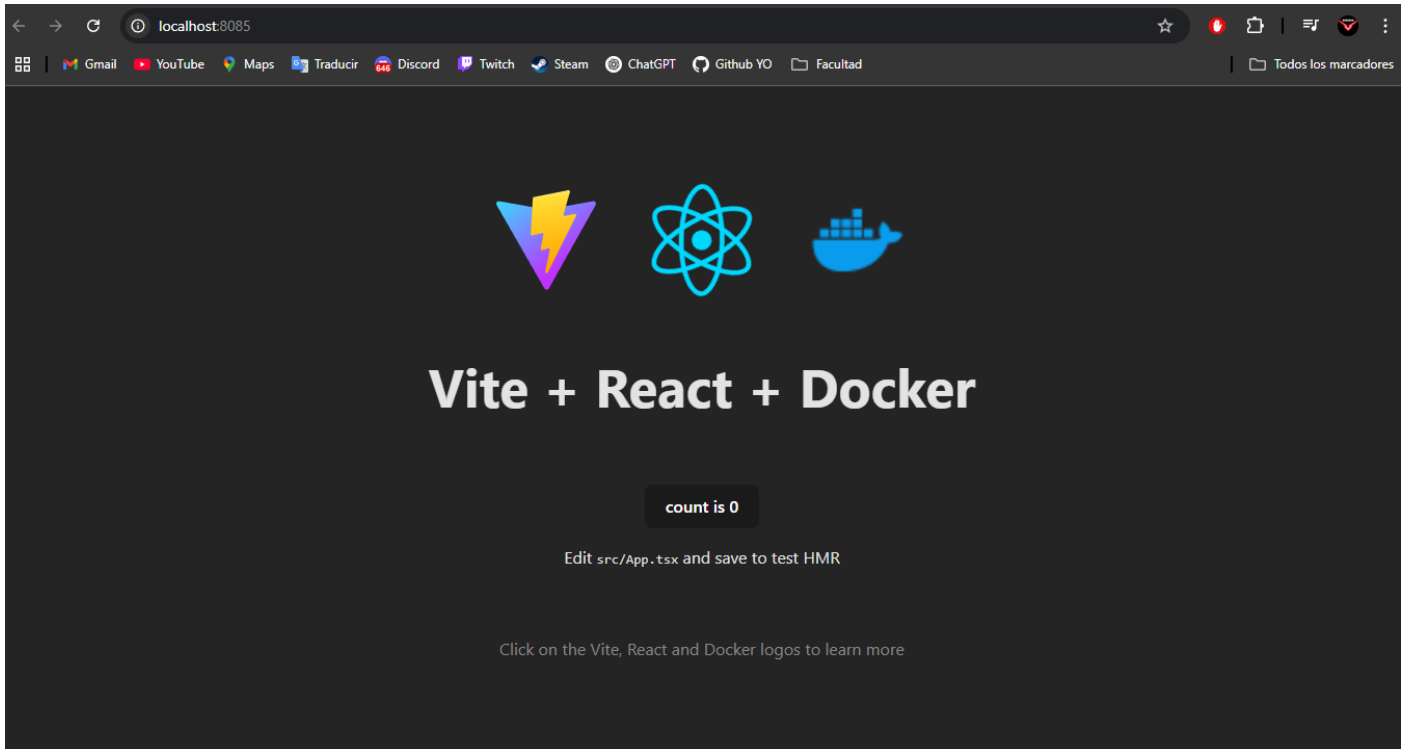
Kubernetes es una plataforma de orquestación de contenedores que permite automatizar el despliegue, escalado y gestión de aplicaciones en contenedores.

Relación con Docker:

- Docker crea y ejecuta contenedores
- Kubernetes los administra a gran escala (clústeres, balanceo, alta disponibilidad, etc.)

PARTE 2 – Práctica técnica

- La app corriendo en <http://localhost:8085/>



- docker ps mostrando el contenedor activo

```
PS D:\YO\Documentos\Facultad\Tecnica\4 semestre\Metodologia Sist\arnold_metodologia\tp_docker+react\docker+react> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                    NAMES
7cb5c7521a29   app-react  "docker-entrypoint.s..." 4 hours ago    Up 4 hours    0.0.0.0:8085->5173/tcp    react-vite-app
```

- docker build mostrando la construcción

```
PS D:\YO\Documentos\Facultad\Tecnica\4 semestre\Metodologia Sist\arnold_metodologia\tp_docker+react\docker+react> docker build -t app-react .
[+] Building 1.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                docker:desktop-linux 0.0s
=> => transferring dockerfile: 426B                                              0.0s
=> [internal] load metadata for docker.io/library/node:20-alpine                 0.7s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 85B                                                  0.0s
=> [1/6] FROM docker.io/library/node:20-alpine@sha256:be56e91681a8ec1bba91e3006039bd228dc797fd984794a3efedab325b36e679 0.0s
=> => resolve docker.io/library/node:20-alpine@sha256:be56e91681a8ec1bba91e3006039bd228dc797fd984794a3efedab325b36e679 0.0s
=> [internal] load build context                                                 0.0s
=> => transferring context: 696B                                                0.0s
=> CACHED [2/6] WORKDIR /app                                                     0.0s
=> CACHED [3/6] COPY package*.json ./                                           0.0s
=> CACHED [4/6] RUN npm install                                                  0.0s
=> CACHED [5/6] COPY . .                                                         0.0s
=> CACHED [6/6] RUN npm run build                                                0.0s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.0s
=> => exporting manifest sha256:9b6ac43b29695144696db0ae6c96937d7ef741347a1eca0ebc475b7fbadb1715 0.0s
=> => exporting config sha256:5a58e1ecac13232dc5304485944e74a261b45544f06ae9e65c9f74a02fcf55a9 0.0s
=> => exporting attestation manifest sha256:98abc0129441b8878b59a28f97d6660e0d1c62ebb82bedd4b437e0270009c256 0.0s
=> => exporting manifest list sha256:e5e39f08de9699b7b8bd322ba04551a332a9ccfec12c16b2b3df843a52394681 0.0s
=> => naming to docker.io/library/app-react:latest                             0.0s
=> => unpacking to docker.io/library/app-react:latest                           0.0s

View build details: docker-desktop:///dashboard/build/desktop-linux/desktop-linux/k6geobe3ocy96a3p6sc4n9hjq
```

Repositorio general:

https://github.com/ZERO-JIFAR/arnold_metodologia

Repositorio del proyecto:

https://github.com/ZERO-JIFAR/arnold_metodologia/tree/main/tp_docker%2Breact/docker%2Breact