

## 1) Conceptos Básicos

### 1.1 Explica en tus propias palabras qué es Docker y cuál es su propósito principal.

Docker es una herramienta que nos ayuda a facilitar la creación de aplicaciones dentro de contenedores. Dentro de los contenedores se tiene todo lo necesario para correr dicha aplicación y lo hace de manera ligera.

Su propósito principal es estandarizar el empaquetamiento de aplicaciones y que estas se puedan correr desde cualquier entorno que este utilizando docker gracias a la virtualización

### 1.2 ¿Cuál es la diferencia entre una imagen y un contenedor en Docker?

Las Imágenes son los paquetes que se necesitan para que una aplicación funcione, pueden ser bibliotecas, dependencias, librerías etc.

Un contenedor es como una caja en la que se almacenan las imágenes que se necesitan para correr una aplicación.

## 2) Instalación y Configuración

R= Lo primero que tenemos que hacer es revisar nuestra computadora y checar si tenemos activado Hyper V , después ver si esta activa la virtualización habilitada en nuestro CPU, posteriormente revisar que tengamos la herramienta de Linux instalándola desde Ubuntu para ahora si irnos a la página de Docker e instalar la versión para Windows , todos estos pasos son para que pueda funcionar dicho programa

### 2.1 Proporciona los pasos básicos para instalar Docker en un sistema operativo WINDOWS/LINUX/MAC (ELIGE UNO).

Primero en un navegador buscamos Docker y nos despliega las opciones, por lo general será la primera opción, seleccionamos en mi caso MAC y elegimos la opción que el sitio nos da y descargamos la aplicación de escritorio

### 2.2 Describe brevemente qué es Docker Hub y cómo se puede utilizar en el contexto de Docker.

Docker Hub es una nube en donde estarán las imágenes de distintos desarrolladores , para poder trabajar en equipo y que estas se puedan conectar entre si , el contexto podría ser el de Cuevaana donde comparten entre desarrolladores las imágenes para poder expandir el catalogo de series y de diferentes tipos de idiomas

## 3) Manipulación de Contenedores

### 3.1 Crea un archivo Dockerfile simple que utilice una imagen base de Ubuntu y ejecute el comando

"Hello, Docker!" al iniciar el contenedor. (PARA ESTE CASO PUEDES USAR CHATGPT O BLACKBOX IA)

FROM ubuntu

CMD echo "Hello Docker"

3.2 Explica la diferencia entre los comandos docker ps y docker ps -a. ¿Qué información proporciona cada uno?

El comando docker ps muestra los contenedores que están corriendo

Mientras docker ps -a muestra todos los contenedores aun cuando no estén corriendo

#### 4) Redes en Docker

4.1 ¿Cómo se pueden listar las redes disponibles en Docker?

docker network ls

4.2 Crea una red en Docker llamada "mi\_red" y explica cómo asignar un contenedor a esta red al momento de iniciarlo.

Primero creamos un archivo docker-compose.yml y le especificamos:

Version

Services:

Nombre del contenedor: (Aqui se pone el nombre del contenedor)

Build:

Ports:

Volumes:

environment:

#### 5) Persistencia de Datos

5.1 Explica la diferencia entre montar un volumen y copiar archivos directamente dentro de un contenedor.

Al montar un volumen los datos serán persistentes, se podrán compartir entre contenedores lo cual hace más eficiente al desarrollar

Mientras que copiar archivos directamente en el contenedor, los datos son estáticos y deben estar empaquetados con la imagen

5.2 ¿Cuál es la ventaja de utilizar volúmenes en Docker para la persistencia de datos?

En que los datos se podrán compartir entre contenedores, los datos se sincronizan automáticamente

#### 6) Composición con Docker Compose

6.1 ¿Qué es Docker Compose y para qué se utiliza?

Es una herramienta que permite gestionar aplicaciones multicontenedor

6.2 Crea un archivo docker-compose.yml para definir dos servicios: uno que utilice la imagen de Mongo y otro que utilice la imagen de Node Version 14. Asegúrate de especificar la red a la que pertenecerán ambos servicios.

Version: "3"

Services:

    Mongodb:

        Image: mongo:latest

        Networks:

        - my-network

    Node\_app:

        Image: node:14

        Networks:

        - my-network

Networks:

    my-networks:

## 7) Resolución de Problemas

7.1 Imagina que un contenedor no se inicia correctamente.

Proporciona algunos pasos que seguirías para identificar y solucionar el problema.

Me paso que tenia mal algunos comandos del archivo Dockerfile, revisaria que estuvieran bien escritos, si ya esta verificado, revisar que los comandos de la terminal estén bien escritos, revisar que el docker desktop este funcionando

7.2 ¿Cómo puedes acceder a la shell de un contenedor en ejecución?

Usando el comando docker exec